

Enterprise SONiC Distribution by Dell Technologies

User Guide Release 4.2.0

Notes, cautions, and warnings

 **NOTE:** A NOTE indicates important information that helps you make better use of your product.

 **CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

 **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

Contents

Chapter 1: SONiC overview.....	11
SONiC and Dell Technologies.....	11
Additional Enterprise SONiC documentation.....	13
Chapter 2: Legacy SONiC configuration models.....	15
Chapter 3: Management Framework	19
Chapter 4: Getting Started.....	20
Log in to SONiC.....	20
CLI basics.....	21
SSH login.....	22
Using a DHCP server to automate the initial configuration.....	23
Default startup configuration.....	24
Show running configuration.....	24
Show configuration.....	26
View software version.....	26
View system hardware.....	29
View core service status.....	34
View system firmware.....	37
Restore factory defaults.....	38
Reset Enterprise SONiC password.....	38
Chapter 5: System management and setup.....	40
Software image management.....	41
Firmware installation.....	44
Install a software patch.....	46
Using USB storage media.....	48
System file management.....	50
Configure user login lockout.....	52
Interface naming modes.....	53
Role-based access control.....	56
Create users and assign roles.....	57
View local and remote users.....	59
Authentication, authorization, and accounting.....	59
Configure authorization.....	59
Configure authentication.....	60
Configure TACACS+ servers.....	62
Configure RADIUS servers.....	64
Dynamic Authorization Server.....	66
Configure LDAP.....	68
Network Time Protocol	73
DNS server.....	78
Fast reboot.....	79

Simple Network Management Protocol.....	79
SNMP versions.....	79
MIBs.....	80
SNMP traps.....	80
Configure SNMP.....	81
Secure SNMP access.....	86
Example: Configure SNMPv2c.....	87
Example: Configure SNMPv3.....	87
Example: Configure agent address for Management VRF.....	88
Example: Read service tag using SNMP.....	88
Dynamic Host Configuration Protocol.....	89
DHCP relay.....	89
DHCP snooping.....	99
DHCP snooping and DHCP relay on same VLAN.....	102
Third party containers.....	103
L2 and L3 switch profiles.....	108
L2/L3 host and route scaling.....	109
Cut-through switching.....	111
Configure CPU polling interval	111
Chapter 6: Zero touch provisioning.....	112
ZTP DHCP options.....	113
ZTP JSON file.....	114
View ZTP status.....	116
ZTP DHCP server configuration.....	117
Additional ZTP JSON configuration objects.....	117
ZTP JSON file plug-ins.....	118
ZTP JSON dynamic content.....	119
Using ZTP plug-ins in JSON configuration objects.....	121
ZTP provisioning using a USB drive.....	128
Chapter 7: Interfaces.....	130
Interface configuration mode.....	130
Basic interface configuration.....	131
Configure access and trunk interfaces.....	138
Interface ranges.....	142
Forward error correction.....	145
Port groups.....	148
S5296F-ON port groups	150
S5248F-ON port groups	150
Port profiles.....	151
Port breakouts.....	153
Port channels.....	156
Port channel configuration.....	157
Port channel reconfiguration.....	160
Port-channel graceful shutdown.....	161
VLANs.....	162
VLAN configuration.....	162
VLAN autostate.....	164

Q-in-Q VLAN tunneling.....	165
Configure Q-in-Q VLAN tunneling.....	167
View Q-in-Q VLAN tunneling	169
VLAN translation.....	170
Configure VLAN translation.....	171
View VLAN translation.....	173
Configure VLAN translation using REST API.....	173
Layer 3 subinterfaces.....	174
Show transceivers.....	175
Media-based port autoconfiguration.....	179
High-power optics.....	179
Enable high-power optics on a port.....	179
Chapter 8: Layer 2.....	181
Link aggregation control protocol.....	181
Link layer discovery protocol.....	182
Media access control.....	184
Spanning-tree protocol.....	187
About STP.....	188
Change STP mode.....	188
Enable or disable STP.....	188
Enable BPDU filtering.....	188
Recover from BPDU guard violations.....	189
Spanning-tree link type for rapid state transitions.....	190
Dynamic path cost calculation.....	190
Rapid per-VLAN spanning-tree.....	190
Rapid spanning-tree protocol.....	194
Multiple spanning tree protocol.....	194
Port monitoring.....	199
Create a mirror session.....	199
Flow-based port monitoring.....	201
Port Security.....	204
Configure port security.....	204
Chapter 9: Layer 2 Edge features.....	206
Port access control.....	207
Power over Ethernet.....	214
LLDP-MED for voice VLANs.....	217
Reserved VLANs.....	219
Multislot External Power Supply	220
Chapter 10: Layer 3.....	222
Configure IPv4 and IPv6 address.....	222
Virtual routing and forwarding.....	223
Management VRF.....	224
Configure nondefault VRF instances.....	224
Border Gateway Protocol.....	225
Enable BGP.....	226
Configure BGP router.....	228

Configure BGP address family.....	232
Route reflection.....	234
Configure BGP neighbors.....	235
BGP peer groups.....	243
BGP routing policy filters.....	249
Unnumbered BGP.....	253
View BGP configuration and operation.....	253
View BGP IPv4 routes.....	256
IPv6 Neighbor Discovery Protocol.....	259
View BGP IPv6 routes.....	263
IPv4 Address Resolution Protocol.....	265
View IPv4 ARP entries.....	266
View IPv6 NDP entries.....	268
Open Shortest Path First.....	269
Enable OSPFv2	270
Enable OSPF on Interfaces	271
Configure OSPF router ID	272
Configure OSPF area level authentication type	272
Configure OSPF interface level authentication type and keys	272
Configure OSPF passive interfaces	273
Configure OSPF virtual links	273
Configure OSPF ABR type	274
Configure OSPF area shortcuts.....	275
Configure OSPF RFC compatibility.....	275
Configure OSPF adjacency logging.....	275
Configure OSPF LSA timers	275
Configure OSPF SPF throttle timers	276
Configure OSPF max-metric advertising	276
Configure OSPF route distances	277
Configure OSPF auto cost reference bandwidth	277
Configure OSPF stub area and its parameters	277
Configure OSPF inter area summary route filters	278
Configure OSPF route redistribution	278
Configure OSPF default route origination	279
Configure OSPF interface parameters	279
OSPF operational data display commands	282
OSPFv2 graceful restart.....	288
Route-maps.....	291
Create a route-map.....	292
Static routes.....	293
View IP routes.....	296
Policy-based routing.....	297
PBR forwarding policies.....	298
Classify PBR traffic	298
Configure and apply PBR forwarding policies.....	299
Example: PBR forwarding configuration	304
View PBR forwarding configuration	304
PBR quick configuration	306
Virtual Router Redundancy Protocol.....	307
VRRP configuration.....	307

Create virtual router.....	308
Group version.....	309
Virtual IP addresses.....	309
Configure virtual IP addresses in a VRF.....	310
Set group priority.....	311
Disable preempt.....	312
Advertisement interval.....	312
Interface tracking.....	313
Network Address Translation	314
Enable NAT.....	315
NAT configuration.....	315
View NAT configuration.....	317
ECMP.....	318
IP helper.....	319
Chapter 11: Multicast.....	323
Configure multicast routing.....	323
Internet group management protocol.....	323
Supported IGMP versions.....	324
Query interval.....	324
Last member query interval.....	324
Response timer.....	325
Select IGMP version.....	325
View IGMP-enabled interfaces and groups.....	325
IGMP snooping.....	326
Protocol-independent multicast.....	328
PIM-SSM.....	329
PIM global configuration commands.....	329
PIM interface-specific commands.....	330
Clear PIM configuration.....	331
View multicast PIM-SSM global configuration.....	332
Sample PIM-SSM configuration.....	337
Chapter 12: VXLAN	341
VXLAN concepts.....	344
VXLAN as NVO solution.....	344
Configure VXLAN.....	345
Configure VTEP.....	345
Configure symmetric IRB.....	346
View symmetric IRB configuration.....	350
Configure asymmetric IRB.....	351
View asymmetric IRB configuration.....	357
VXLAN statistics.....	358
Configure EVPN.....	358
Filter Type-2, Type-3, and Type-5 EVPN routes.....	366
Multi-site data center interconnect.....	367
EVPN multihoming.....	372
Chapter 13: Multichassis LAG.....	383

MCLAG configuration.....	384
View MCLAG configuration.....	389
MCLAG peer gateway.....	390
Troubleshoot MCLAG.....	391
Chapter 14: Access control lists.....	394
ACLs for security and packet classification.....	394
Configure ACLs.....	395
L2 and L3 ACL interaction.....	397
Security ACL examples.....	398
ACLs for flow-based services.....	398
Configure flow-based ACLs	399
Flow-based ACL examples.....	401
Policy-based replication groups	402
ACL consistency checker.....	404
Enable ACL counters.....	405
View ACL configuration and counters.....	406
Chapter 15: Quality of Service.....	410
Flow-based QoS.....	411
Classify traffic using modular ACLs.....	411
Configure and apply QoS policies.....	412
Example: Flow-based QoS	414
WRED and ECN.....	415
Scheduler policy.....	417
Port shaping.....	418
QoS maps.....	419
DSCP to traffic class-map.....	419
Dot1p to traffic class-map.....	419
Traffic class to queue map.....	420
Traffic class to priority group map.....	420
Traffic class to dot1p map.....	421
Traffic class to DSCP map.....	421
PFC priority to queue map.....	422
View QoS configuration.....	422
Control plane policing.....	424
Priority flow control.....	431
Storm control.....	435
Buffer management.....	435
Pre-configured NPU defaults.....	438
Pre-configured lossless buffers.....	438
Configure buffers.....	441
Configure ingress buffer.....	443
Configure egress buffer.....	443
Chapter 16: RDMA over Converged Ethernet	444
Enable RoCEv2 with default configuration	446
View RoCEv2 default configurations	446
Reconfigure RoCEv2 default configurations	452

RoCE traffic hashing	454
Chapter 17: Switch protection.....	456
Bidirectional Forwarding Detection.....	456
BFD session states.....	457
Three-way handshake.....	457
BFD configuration.....	458
Configure BFD.....	458
BFD for BGP.....	459
BFD for OSPF.....	460
BFD for PIM.....	460
BFD profile.....	461
Link state tracking.....	467
Link state tracking configuration notes.....	468
Unidirectional link detection.....	468
Configure UDLD.....	470
View UDLD configuration.....	471
Link-error port disabling.....	472
Chapter 18: sFlow.....	474
Configure sFlow.....	474
View sFlow statistics.....	476
sFlow configuration example.....	476
Chapter 19: REST API.....	478
REST API authentication.....	479
View REST API authentication.....	485
REST API requests using curl.....	488
YANG PATCH operation.....	489
REST API examples.....	490
Chapter 20: gRPC Network Management Interface.....	498
gNMI certificate authentication.....	500
gNMI JWT authentication.....	504
gNMI password authentication.....	505
View gNMI authentication.....	506
gNMI request examples.....	507
gNMI for streaming telemetry.....	519
gRPC network operations interface.....	522
Chapter 21: Using OpenConfig paths.....	524
Chapter 22: Basic troubleshooting.....	528
System log.....	532
Audit log.....	541
Subscribe to Event messages using gNMI.....	542
Chassis locator LED.....	542
Port locator LED.....	543
Ping.....	544

Traceroute.....	546
Enable core file generation.....	547
View core file information.....	547
Error disable recovery.....	548
Using port LEDs.....	549
Port up or down troubleshooting.....	549
Physical link signal.....	551
Investigating packet drops.....	552
Configure packet drop counters.....	552
Buffer thresholds to detect congestion.....	556
Debug application for congestion and drops.....	559
Isolate SONiC switch from network.....	567
NAT troubleshooting.....	567
Kernel dump.....	569
Check memory usage.....	571
View memory usage for processes.....	572
View memory usage for dockers.....	573
View system memory usage.....	574
View the reason for down interfaces.....	576
System reboot reason.....	579
Unreliable Loss of Signal.....	580
Transceiver and cable diagnostics.....	581
Chapter 23: Advanced troubleshooting with Telemetry and Monitoring.....	587
Inband flow analytics.....	587
Packet drop monitoring.....	591
Tail stamping.....	595
Chapter 24: Support resources.....	597
Appendix A: MIB objects.....	598

SONiC overview

What is SONiC

SONiC is an open-source, Linux-based network operating system (NOS) that runs on switches from multiple vendors and ASICs. SONiC stands for Software for Open Networking in the Cloud. It implements standard Layer 2 and Layer 3 protocols, and provides developers with a straightforward way to add new features.

SONiC offers teams the flexibility to create data center networking solutions, while using the collective strength of a large ecosystem with an active developer community. SONiC is designed for scalability and is in production today in large data center fabrics. Some of SONiC's benefits include:

- Hardware independence
- Containerized architecture
- Open-sourced
- Access to a growing community

For more information, see [What is SONiC?](#).

Topics:

- [SONiC and Dell Technologies](#)
- [Additional Enterprise SONiC documentation](#)

SONiC and Dell Technologies

Dell Technologies introduces the Enterprise SONiC operating system as a hardened, validated, and supported version of SONiC for switch configuration and monitoring. It includes distribution of open-source community SONiC, and additional features to support the ecosystem and partners.

Enterprise SONiC supports an intuitive command-line interface, and object-based administration through a REST interface and Google's gRPC Network Management Interface (gNMI).

Enterprise SONiC Distribution by Dell Technologies

Enterprise SONiC is offered in the following bundles. Customers can deploy the most appropriate bundle for their network requirements:

- Cloud Standard
- Cloud Premium
- Enterprise Standard
- Enterprise Premium
- Edge Standard

(i) **NOTE:** The Edge Standard bundle is supported only on N3248TE-ON, N3248PXE-ON, N3248X-ON, E3248P-ON, and E3248PXE-ON switches. These Dell PowerSwitch platforms cannot run the Cloud and Enterprise bundles. All other Dell PowerSwitch platforms cannot run the Edge bundle.

Dell PowerSwitch platforms

Enterprise SONiC is supported on the following ONIE-enabled Dell PowerSwitch platforms (which may have a different SONiC network operating system (NOS) or OS10 installed) with the specified port speeds:

- Yes means Supported; No means Not Supported. Default means default speed at access (user) ports.
- Dell PowerSwitches that are marked with * (asterisk) support Power over Ethernet (PoE).

Table 1. Enterprise SONiC on Dell PowerSwitches

Dell PowerSwitch	400G	200G	100G	50G	40G	25G	10G	5G	2.5G	1G	100M	10M
Z9664F-ON	Default	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes	No	No

Table 1. Enterprise SONiC on Dell PowerSwitches (continued)

Dell PowerSwitch	400G	200G	100G	50G	40G	25G	10G	5G	2.5G	1G	100M	10M
Z9432F-ON	Default	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes	No	No
Z9332F-ON	Default	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes	No	No
Z9264F-ON	No	No	Default	No	Yes	Yes	Yes	No	No	Yes	No	No
S5448F-ON	Yes	Yes	Default	No	Yes	Yes	Yes	No	No	Yes	No	No
S5296F-ON	No	No	Yes	Yes	Yes	Default	Yes	No	No	Yes	No	No
S5248F-ON	No	No	Yes	No	Yes	Default	Yes	No	No	Yes	No	No
S5232F-ON	No	No	Default	Yes	Yes	Yes	Yes	No	No	Yes	No	No
S5224F-ON	No	No	Yes	No	Yes	Default	Yes	No	No	Yes	No	No
S52212-ON	No	No	Yes	No	Yes	Default	Yes	No	No	Yes	No	No
N3248PXE-ON*	No	No	Yes	No	Yes	Yes	Default	Yes	Yes	Yes	Yes	Yes
N3248X-ON	No	No	Yes	No	Yes	Yes	Default	Yes	Yes	Yes	Yes	Yes
N3248TE-ON	No	No	Yes	No	Yes	No	Yes	No	No	Default	Yes	Yes
E3248PXE-ON*	No	No	Yes	No	Yes	Yes	Default	Yes	Yes	Yes	Yes	Yes
E3248P-ON*	No	No	Yes	No	Yes	No	Yes	No	No	Default	Yes	Yes

(i) NOTE: For information on the breakout modes, cables, and optics supported on each switch, contact your designated sales representative.

(i) NOTE: The Z9100-ON is supported in 3.3 and earlier releases; it is not supported in the 4.1.0 release.

(i) NOTE: To install Enterprise SONiC on an ONIE-enabled Dell PowerSwitch that is running OS10 or another SONiC NOS and reboot the switch, use ZTP — see [Zero touch provisioning](#). For more information, see [Third-Party NOS Installation Guide for Dell EMC PowerSwitch Data Center Switches](#).

Table 2. Dell PowerSwitches supported in each bundle

Enterprise SONiC bundle	N-Series: N3248PXE-ON, N3248TE-ON, N3248X-ON	S-Series: S5296F-ON, S5232F-ON, S5248F-ON, S5224F-ON, S5212F-ON, S5448F-ON	Z-Series: Z9264F-ON, Z9332F-ON, Z9432F-ON, Z9664F-ON	E-Series: E3248PXE- ON, E3248P-ON
Cloud Standard	Not Supported	Supported	Supported	Not Supported
Cloud Premium	Not Supported	Supported	Supported	Not Supported
Enterprise Standard	Not Supported	Supported	Supported	Not Supported
Enterprise Premium	Not Supported	Supported	Supported	Not Supported
Edge Standard	Supported	Not Supported	Not Supported	Supported

Enterprise SONiC features in bundles

All software features described in the Enterprise SONiC User Guide are available in all Dell PowerSwitch bundles, except for the following features:

Table 3. Features not available in all bundles

Enterprise SONiC feature	Cloud Standard	Cloud Premium	Enterprise Standard	Enterprise Premium	Edge Standard
Buffer thresholds	Supported	Supported	Supported	Supported	Not Supported
Debug application	Not Supported	Not Supported	Not Supported	Supported	Not Supported
Dot1x authentication	Not Supported	Not Supported	Not Supported	Not Supported	Supported

Table 3. Features not available in all bundles (continued)

Enterprise SONiC feature	Cloud Standard	Cloud Premium	Enterprise Standard	Enterprise Premium	Edge Standard
Dynamic ACLs	Not Supported	Not Supported	Not Supported	Not Supported	Supported
EVPN	Not Supported	Not Supported	Supported	Supported	Supported
EVPN Multihoming	Not Supported	Not Supported	Supported	Supported	Supported
gNMI	Supported	Supported	Supported	Supported	Supported only on E-Series
Guest VLANs	Not Supported	Not Supported	Not Supported	Not Supported	Supported
IGMP and IGMP snooping	Not Supported	Not Supported	Supported	Supported	Supported
Inband flow analytics (IFA)	Not Supported	Supported	Not Supported	Supported	Not Supported
IPv4 PIM source-specific multicast (SSM)	Not Supported	Not Supported	Supported	Supported	Supported
MAC authentication bypass	Not Supported	Not Supported	Not Supported	Not Supported	Supported
Multiple Spanning Tree Protocol (MSTP)	Not Supported	Not Supported	Supported	Supported	Supported
Multisite data center interconnect	Not Supported	Not Supported	Supported	Supported	Not Supported
Network Address Translation (NAT)	Supported	Supported	Supported	Supported	Not Supported
Packet drop monitoring	Not Supported	Not Supported	Not Supported	Supported	Not Supported
Power over Ethernet (PoE)	Not Supported	Not Supported	Not Supported	Not Supported	Supported
PVST and RPVST+	Not Supported	Not Supported	Supported	Supported	Supported
Q-in-Q VLAN tunneling	Not Supported	Not Supported	Supported	Supported	Not Supported
RADIUS-provided VLANs	Not Supported	Not Supported	Not Supported	Not Supported	Supported
Reserved VLANs	Supported	Supported	Supported	Supported	Supported
Tail stamping	Not Supported	Supported	Not Supported	Supported	Not Supported
Unauthenticated VLANs	Not Supported	Not Supported	Not Supported	Not Supported	Supported
VLAN translation	Not Supported	Not Supported	Supported	Supported	Not Supported
Voice VLANs	Not Supported	Not Supported	Not Supported	Not Supported	Supported
VXLAN	Not Supported	Not Supported	Supported	Supported	Supported

 **NOTE:** For detailed information about the individual features supported on each Enterprise SONiC switch, see the **Standard and premium feature matrix** section in the *Dell Enterprise SONiC Distribution Compatibility Matrix*.

Additional Enterprise SONiC documentation

Table 4. Additional documentation

Document	Description
<i>Enterprise SONiC Distribution by Dell Technologies, 4.2.0 Quick Start Guide</i>	Installation and initial setup tasks

Table 4. Additional documentation (continued)

Document	Description
<i>Enterprise SONiC Distribution by Dell Technologies, 4.2.0 Management Framework CLI Reference Guide</i>	Management Framework CLI command syntaxes and examples
<i>Enterprise SONiC Distribution by Dell Technologies, 4.2.0 Compatibility Matrix</i>	Supported software features, scalability, interfaces, breakouts, cables, and optics
<i>Enterprise SONiC Distribution by Dell Technologies, 4.2.0 High Power Optics</i>	Supported high-power optics, thresholds, and switch ports
<i>Enterprise SONiC Distribution by Dell Technologies, 4.2.0 Release Notes</i>	New features introduced in the release; known and fixed issues

 **NOTE:** This guide may contain language from third-party content that is not under Dell Technology control and is not consistent with Dell Technology guidelines. Dell Technology plans to update this reference guide when the third party updates their content.

Legacy SONiC configuration models

There are various management models available to manage, configure, and administer a SONiC-based network operating system (NOS). However, in practice, these models become complicated and difficult to use for network administrators:

- Legacy SONiC configuration models are different and not centralized — An administrator has to navigate between different shells, such as Linux, the legacy SONiC CLI and FRR shell, JSON files and .xml files, to configure a switch.
- There are several sources of truth for configuration backups — An administrator cannot back up a switch configuration from a single place.
- Root privilege is often required to run configuration commands — An administrator needs root privilege to run configuration commands in the legacy SONiC CLI and many Linux commands in the Linux shell.

To avoid this complexity, Dell Technologies pioneered the SONiC Management Framework, which provides a unified model to manage and configure the features available on Enterprise SONiC as a "single source of truth". To improve the user experience of SONiC-based NOSs, Dell Technologies contributed the Management Framework to the SONiC community. For more information, see [Management Framework](#). This chapter describes the main legacy SONiC configuration models.

 **NOTE:** Dell Technologies strongly recommends that you only use the Management Framework to configure Enterprise SONiC features. New features and functionality in Enterprise SONiC will be available only in the Management Framework.

Linux shell

When you log in to the SONiC NOS as an administrator or sudo user (for example, admin), you are placed in the Linux shell.

```
admin@10.10.10.10's password:
Linux sonic 5.10.0-8-2-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
You are on

/   \ /   \ \   \ \   \ /   \
\   / |   | |   | |   | |   |
 )   | |   | |   | |   | |   |
 |   / \   / | \   | \   | \   |
-- Software for Open Networking in the Cloud --
Unauthorized access and/or use are prohibited.
All access and/or use are subject to monitoring.

Help:      http://azure.github.io/SONiC/
Last login: Wed Feb 22 18:20:41 2023 from 100.64.52.123
admin@sonic:~$
```

To display the contents of a folder:

```
admin@sonic:~$ sudo ls /etc/sonic/
agent_config.cfg          frr                  snmp.yml
asic_config_checksum       generated_services.conf  sonic_branding.yml
config_db.json             hamd                sonic_version.yml
constants.yml              init_cfg.json        updategraph.conf
```

To access the Management Framework CLI, enter `sonic-cli` from the Linux shell. You are placed at the command-line interface in Exec mode (`sonic#`). From this mode, you can run configuration commands to provision the switch or show commands to monitor various switch functions. Use `?` to display the available commands.

 **NOTE:** You cannot run the `sonic-cli` command from the root user level.

```
admin@sonic:~$ sonic-cli
sonic# ?
```

clear	Clear commands
configure	Enter configuration mode
consistency-check	Performs consistency check
copy	Perform file operations
debug	Enter debugsh mode
delete	Delete the file from local filesystem
dir	Show folder contents
exit	Exit from the CLI
fast-reboot	fast-reboot
hardware	ASIC parameters related commands
image	Image related commands
interface	interfaces Utility
locator-led	Locator Chassis LED Utility
logger	Enter messages into the system log
ls	Show folder contents
no	No commands under Exec mode
ping	Send ICMP ECHO_REQUEST to network hosts
ping6	Send ICMPv6 ECHO_REQUEST to network hosts
poe	Reset POE Port(s)
reboot	reboot
renew	Renew commands
show	Display running system information
terminal	Set terminal settings
test	Run diagnostics test program
tpcm	SONiC image installation manager
traceroute	Print the route packets take to the host
traceroute6	Print the route packets take to the IPv6 host
warm-reboot	warm-reboot
write	Save config

For information about how to use the Management Framework CLI, see [CLI basics](#).

Legacy SONiC CLI

The legacy SONiC CLI is a set of user-space Linux applications, such as `config` and `show` that allow you to configure and provision SONiC switches. This CLI uses the Python Click library. The Click library provides developers with a customizable approach to create command-line tools, perform various configurations, and leverage the defaults that are provided with the package. For more information about the Click Python package, see [Welcome to Click](#) at <https://click.palletsprojects.com/en/7.x/>.

To display the available configuration options supported by the legacy SONiC CLI, enter `sudo config ?` from the Linux shell. Root privilege is required to run the configuration and show commands.

```
admin@sonic:~$ sudo config ?
Usage: config [OPTIONS] COMMAND [ARGS]...

SONiC command line - 'config' command

Options:
  --help  Show this message and exit.

Commands:
  aaa          AAA command line
  acl          ACL-related configuration tasks
  bgp          BGP-related configuration tasks
  classifier   Classifiers related configuration tasks
  copp         Configure COPP
  core         Configure coredump
  custom_assert Configuration action on assert
  ecn          ECN-related configuration tasks
  export        Flow related configuration tasks
  flow          Configure hardware parameters
  hardware     igmp-snooping configuration tasks
  hostname     Interface-related configuration tasks
  igmp_snooping
  interface...
  ...
```

To get help at the command level, enter ? in a command syntax; for example:

```
admin@sonic:~$ sudo config ztp ?
Usage: config ztp [OPTIONS] COMMAND [ARGS]...

    Configure Zero Touch Provisioning

Options:
    -?, -h, --help  Show this message and exit.

Commands:
    disable  Administratively Disable ZTP.
    enable   Administratively Enable ZTP.
    run      Restart ZTP of the device.
```

To save configuration changes made with the legacy SONiC CLI , enter the config save command:

```
admin@sonic:~$ sudo config save
Existing file will be overwritten, continue? [y/N]: y
Running command: /usr/local/bin/sonic-cfggen -d --print-data > /etc/sonic/config_db.json
admin@sonic:~$
```

Enter show commands to display switch configuration; for example:

```
admin@sonic:~$ sudo show ztp
ZTP Admin Mode : False
ZTP Service     : Inactive
ZTP Status      : Not Started

ZTP Service is not running
```

FRR shell CLI

To configure routing, you can also use the open-source package, Free Range Routing (FRR), through its FRRouting shell. FRR provides a suite of standard IP routing protocols, including BGP, RIP, OSPF, EIGRP, BABEL, IS-IS, PIM, IGMP, and VRRP. In FRR, each routing protocol operates independently with a separate daemon to ensure high resiliency. For more information about FRR, see [FRR Overview](#) at <https://docs.frrouting.org/en/latest/overview.html> and [FRRouting initialization and configuration](#).

Enterprise SONiC has qualified a subset of FRR features. For information about the supported FRR features, see the *Enterprise SONiC Compatibility Matrix*.

To enter the FRR shell from SONiC, use the vtysh command.

```
admin@sonic:~$ vtysh
Hello, this is FRRouting (version 7.2-sonic).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

sonic#
```

Enter routing configuration commands at the prompt. For example, using the FRR shell, you can configure eBGP with these commands:

```
sonic# configure
sonic(config)# router bgp 1234
sonic(conf-router)# bgp router-id 10.0.2.1
sonic(conf-router)# bgp graceful-restart
sonic(conf-router)# bgp bestpath as-path multipath-relax
sonic(conf-router)# neighbor SPINE peer-group
sonic(conf-router)# neighbor SPINE timers 3 9
sonic(conf-router)# neighbor SPINE advertisement-interval 5
sonic(conf-router)# neighbor 192.168.1.0 peer-group SPINE
sonic(conf-router)# neighbor 192.168.1.0 remote-as 65100
sonic(conf-router)# neighbor 192.168.2.0 peer-group SPINE
sonic(conf-router)# neighbor 192.168.2.0 remote-as 65100
sonic(conf-router)# address-family ipv4 unicast
```

```
sonic(conf-router-af) # exit-address-family  
sonic(conf-router) # exit  
sonic(config)#
```

i | NOTE: FRR configurations using VTYSH are lost during an Enterprise SONiC upgrade unless you set the routing mode to `split`, save the setting, and reboot the device before entering the configuration changes.

FRR operating modes

i | NOTE: The Management Framework is compatible only with FRR separated mode; it is not compatible with other FRR operating modes. If you plan to use them, do not use the Management Framework.

Enterprise SONiC operates in an FRR mode called *separated*. In this mode, FRR expects the configuration of its individual modules to be provided in separate configuration files. The Management Framework leverages separated mode to centralize all FRR operations in the `config_db.json` file, and keep SONiC databases synchronized so that users do not need to maintain individual FRR configuration files.

On an Enterprise SONiC switch, separated FRR mode is the default. To view the FRR operating mode, enter the command:

```
admin@sonic:~$ sudo grep routing_config_mode /etc/sonic/config_db.json  
"docker_routing_config_mode": "separated"
```

If an empty result is returned, the FRR operating mode is *separated*.

i | NOTE: Dell Technologies strongly recommends that you use FRR in separated mode and use only the Management Framework to configure Enterprise SONiC features. New features and functionality in Enterprise SONiC will be available only in the Management Framework.

i | NOTE: During an Enterprise SONiC upgrade, FRR configurations made using VTYSH commands will be lost unless you set the `docker_routing_config_mode` to `split`, save the change, and reload the switch before the VTYSH configurations are applied in the upgrade.

Management Framework

When configuring SONiC, a common challenge was navigating between different shells, and different JSON and XML files to configure the switch. Different configuration commands and syntaxes are used across shells, such as Linux and FRR. As a result, the likelihood of configuration errors increases. Also, it is not possible to back up the switch configuration from a single place, requiring multiple steps for system upgrades.

The Management Framework is introduced to resolve these challenges. The Management Framework is a SONiC application that provides a single, consistent, holistic way to validate, configure, and manage the features running on the SONiC operating system. The Management Framework consists of these user interfaces:

Management Framework CLI	Next generation, intuitive command-line interface for IT administrators that supports centralized switch operation and management, and augments the existing SONiC CLIs: <ul style="list-style-type: none">Linux shell commands — Configuration commands that you enter in the Linux shell, such as <code>useradd</code>, <code>ip</code>, and <code>ifconfig</code>.SONiC CLI — The existing legacy CLI that is based on a Python library.Free Range Routing shell CLI — Configuration commands for routing protocols, such as BGP, OSPF, and EVPN. Free Range Routing is also known as <i>FRR</i> and <i>FRRouting</i>.
Programmatic northbound interfaces	In addition to using the Management Framework command-line interface, you can also use REST API and gNMI interfaces that access YANG-modeled data. The Management Framework supports both standards-based data models, such as OpenConfig, and SONiC YANG models for switch configuration. <ul style="list-style-type: none">REST API — See REST API.gRPC Network Management interface (gNMI) — See gRPC Network Management Interface.gRPC Network Operations Interface (gNOI) — See gRPC Network Management Interface.

The Management Framework supports standard and custom YANG models for communication with management systems. The Management Framework runs in a single container called `sonic-mgmt-framework`.

To start the Management Framework CLI, go to [Getting Started](#).

 **NOTE:** Dell Technologies strongly recommends that you use only the Management Framework to configure Enterprise SONiC features. New features and functionality in Enterprise SONiC will be available only in the Management Framework.

Getting Started

After you log in to a switch, you can start using the Management Framework CLI to configure and monitor the SONiC device.

For information about how to use the REST API or gNMI, see [REST API](#) and [gNMNI Network Management Interface](#).

Topics:

- Log in to SONiC
- CLI basics
- SSH login
- Using a DHCP server to automate the initial configuration
- Default startup configuration
- Show running configuration
- Show configuration
- View software version
- View system hardware
- View core service status
- View system firmware
- Restore factory defaults
- Reset Enterprise SONiC password

Log in to SONiC

All SONiC devices support both a serial console-based and SSH-based login. If you use an SSH login, log in to the Management interface (Management0) IP address (see [SSH login](#)).

(i) **NOTE:** Enterprise SONiC uses SSHv2 as its connection protocol for secure connectivity over a network. In this document, references to "SSH" mean "SSHv2".

1. Log in to the Linux shell from the console or through an SSH connection using the default username `admin` and password `YourPaSsWoRd`.

```
At Console:
Debian GNU/Linux 9 sonic tty$1

sonic login: admin
Password: YourPaSsWoRd

SSH from any remote server to sonic can be done by connecting to the IP address of
the Management interface
user@debug:~$ ssh admin@sonic_ip_address(or SONiC DNS Name)
admin@sonic's password:
```

2. When you log in for the first time, you are prompted to change your password. Follow the steps to change your password and log in to the Linux shell. If this is not your first login, go to Step 3.

```
You are required to change your password immediately (root enforced)
Changing password for admin.
(current) UNIX password: YourPaSsWoRd
Enter new UNIX password: ****
Retype new UNIX password: ****
Linux sonic 4.9.0-11-2-amd64 #1 SMP Debian 4.9.189-3+deb9u2 (2019-11-11) x86_64
You are on
/ _ _ _ | / _ _ \ | \ _ | \ _ /| _ _ |
\ _ _ \ | | | | | \ | | | | | |
| _ _ ) | | | | | | \ | | | | | |
| _ _ / \ _ / | | | | | | | | | | | |
```

```
-- Software for Open Networking in the Cloud --
Unauthorized access and/or use are prohibited.
All access and/or use are subject to monitoring.

Help:    http://azure.github.io/SONiC/
admin@sonic:~$
```

3. Enter sonic-cli to access the Management Framework CLI in EXEC mode. From this mode, you can run show commands to monitor various functions on the switch or debug commands to troubleshoot switch operation. Use ? to display the available commands.

```
admin@sonic:~$ sonic-cli
sonic# ?
  clear      Clear commands
  configure   Enter configuration mode
  copy        Perform file copy operations
  debug       Debug commands
  exit        Exit from the CLI
  image       Image related commands
  no          No commands under exec mode
  show        Show running system information
  system     System command
  write       Save config
  ...
```

4. Access CONFIGURATION mode to configure switch settings. In CONFIGURATION mode, you can change the current running configuration. Configuration changes are not automatically saved by default. To save configuration changes, use the write memory command in EXEC mode (see [CLI basics](#)).

```
sonic# config terminal
sonic(config) #
```

To return to the Linux shell:

```
sonic(config)# exit
sonic# exit
admin@sonic:~$
```

For information about how to automate SONiC switch deployment using ZTP with common configuration templates, see [Zero touch provisioning](#).

CLI basics

This information describes how to use the Management Framework CLI from the console or through a network connection to configure and monitor a SONiC device. The Management Framework CLI runs on top of a Linux-based operating system kernel.

CLI command modes

The Management Framework CLI has two top-level modes:

- EXEC mode — Monitor, troubleshoot, check status, and network connectivity
- CONFIGURATION mode — Configure network devices

(i) **NOTE:** When you enter CONFIGURATION mode, you are changing the current running configuration. By default, configuration changes are not automatically saved. To save changes, you must enter the write memory command in EXEC mode.

CLI command hierarchy

Management Framework CLI commands are organized in a hierarchy, which you step through to configure the switch. To move up one command mode, enter the `exit` command. To move directly to the EXEC mode from any submode, enter the `end` command.

```
sonic# config terminal
sonic(config)# interface Eth 1/1
sonic(conf-if-Eth1/1)# no shutdown
sonic(conf-if-Eth1/1)# exit
sonic(config)# exit
sonic# write memory
sonic#
```

OR

```
sonic# config terminal
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# no shutdown
sonic(conf-if-Eth1/1)# end
sonic# write memory
sonic#
```

Management Framework CLI features

Consistent command names	Commands that start the same type of function have the same name. For example, <code>show</code> commands display hardware or software information and statistics; <code>clear</code> commands erase various types of system information, such as interlace counters, ARP entries, and BGP sessions.
Available commands	Information about available commands is provided at each level of the CLI command hierarchy. To view a list of available commands, along with a short description of each command, enter a question mark (?) in the command-line interface.
Command completion	Command completion for command names (keywords) and for command options is available. To complete a command or option that you have partially entered, press the Tab key or the Spacebar . If the partially entered letters are a string that uniquely identifies a command, the complete command name appears. A beep indicates that you have entered an ambiguous command, and the possible completions display. Completion also applies to other strings, such as interface names and configuration statements.
Filtering show output	You can filter <code>show</code> command output to view specific information, or start the command output at the first instance of a regular expression. To filter <code>show</code> command output, enter a vertical bar followed by any of these commands after the <code>show</code> command syntax: <ul style="list-style-type: none">• <code>except</code> — Displays only text that does not match a pattern.• <code>find</code> — Searches for the first occurrence of a pattern and displays all further matches.• <code>grep</code> — Displays only the text that matches a specified pattern.• <code>no-more</code> — Does not paginate output.• <code>save</code> — Saves the output to a file in the home folder of the user. For example, to display only the VLANs configured for port channel 5:

```
sonic# show Vlan | grep PortChannel5
201      Inactive    A  PortChannel5
202      Inactive    T  PortChannel5
203      Inactive    T  PortChannel5
204      Inactive    T  PortChannel5
```

SSH login

By default, SONiC uses DHCP to obtain an IP address for the Management interface (`Management0`) from a DHCP server (see [Using a DHCP server](#)).

 **NOTE:** Enterprise SONiC uses SSHv2 as its connection protocol for secure connectivity over a network. In this document, references to "SSH" mean "SSHv2".

To set up a remote SSH login to configure the switch:

1. Log in to the switch. If you did not already change your password after login, the default credentials are username `admin` and password `YourPaSsWoRd`.
2. Disable the DHCP client, and configure an IP address on the Management interface. Configure a Management route for remote access.

```
sonic# config terminal
sonic(config)# interface Management 0
sonic(conf-if-Management0)# ip address 10.1.1.10/24 gwaddr 10.1.1.1
sonic(conf-if-Management0)# no shutdown
sonic(conf-if-Management0)# exit
sonic(config)# exit
sonic# write memory
```

3. Log in to the Management interface (Management0) IP address in an SSH session. The Management interface must be UP and have an IP address.

```
At Console:
Debian GNU/Linux 9 sonic ttyS1

sonic login: admin
Password: YourPaSsWoRd

SSH from any remote server to sonic can be done by connecting to SONiC IP
user@debug:~$ ssh admin@sonic_ip_address(or SONiC DNS Name)
admin@sonic's password:
```

- When you log in with an `admin` role, you are placed in the Linux shell. The prompt is `admin@sonic:~$`. To access the Management Framework CLI, enter the `sonic-cli` command (see [Log in to SONiC](#)).
- When you log in with an `operator` role, you are placed in the Management Framework CLI. The prompt is `sonic#`. An `operator` user cannot access the Linux shell.

For more information, see [Role-based access control](#).

Using a DHCP server to automate the initial configuration

When you use a DHCP server to boot SONiC, you can load the startup configuration from the `config_db.json` file that is stored on a remote server. The Management interface uses DHCP by default to obtain the Management interface IP address from a DHCP server.

If you use DHCP, you must first configure the DHCP server to provide the necessary information, such as management interface IP address, default route, configuration file name, and the server IP address from which the configuration file is downloaded. SONiC contacts the remote server, downloads the `config_db.json` file, and loads the configuration in the file.

Zero touch provisioning (ZTP) is enabled by default to automate SONiC switch configuration. ZTP uses a DHCP server to download and install a SONiC image and the startup `config_db.json` file. ZTP supports HTTP, TFTP, and FTP to download files, such as `ztp.json`, `config_db.json`, software images, and scripts. For more information, see [ZTP DHCP server configuration](#).

(i) NOTE: The Management IP address that is obtained from a DHCP server is automatically renewed when the DHCP lease expires. The amount of lease time is configured on the DHCP server. To manually renew the Management address before the lease expires, use the `renew dhcp-lease interface Management 0` command.

```
sonic# renew dhcp-lease interface Management 0
```

Default startup configuration

When you install SONiC, if you do not configure switch settings after the switch reboots, the default startup configuration is loaded as the running configuration. See the *Enterprise SONiC Distribution by Dell Technologies Quick Start Guide* for complete information.

On a SONiC switch, the startup configuration is stored in the `/etc/sonic/config_db.json` file if ZTP is disabled. These keys are configured by default in the `config_db.json` file.

```
1) DEVICE_METADATA
2) MAP_PFC_PRIORITY_TO_QUEUE
3) QUEUE
4) PORT
5) CRM
6) PORT_QOS_MAP
7) NTP_SERVER
8) BUFFER_QUEUE
9) WRED_PROFILE
10) TC_TO_PRIORITY_GROUP_MAP
11) BUFFER_PROFILE
12) DEVICE_NEIGHBOR
13) DSCP_TO_TC_MAP
14) TC_TO_QUEUE_MAP
15) CABLE_LENGTH
16) SCHEDULER
17) BUFFER_POOL
```

Show running configuration

To display the running configuration, use the `show running-configuration` command. You can specify the running configuration for a specified interface or software feature.

```
show running-configuration [bfd | bgp | class-map | interface interface-type interface-number | ip | ipv6 | link | mac | mclag | mirror-session | ospf | policy-map | route-map | spanning-tree | tam]
```

- (Optional) `interface interface-type interface-number` — View the configured settings of the specified interface type and number:
 - `interface Eth slot/port[/breakout-port]`
 - `interface PortChannel portchannel-number`
 - `interface Loopback number`
 - `interface Management 0`
 - `interface Vlan vlan-id`
 - `interface vxlan vni, where vni is the VXLAN ID.`

View complete running configuration on switch

```
sonic# show running-configuration
!
ip load-share hash ipv4 ipv4-dst-ip
ip load-share hash ipv4 ipv4-src-ip
ip load-share hash ipv4 ipv4-ip-proto
ip load-share hash ipv4 ipv4-l4-src-port
ip load-share hash ipv4 ipv4-l4-dst-port
ip load-share hash ipv6 ipv6-dst-ip
ip load-share hash ipv6 ipv6-src-ip
ip load-share hash ipv6 ipv6-next-hdr
ip load-share hash ipv6 ipv6-l4-src-port
ip load-share hash ipv6 ipv6-l4-dst-port
hostname sonic

mac address-table aging-time 600
kdump enable
kdump memory 0M-2G:256M,2G-4G:256M,4G-8G:384M,8G-:448M
kdump num-dumps 3
```

```

core enable
factory default profile 13 confirm
ip arp timeout 1800
ipv6 nd cache expire 1800
!
!
qos scheduler-policy copp-scheduler-policy
!
queue 0
  type wrr
  weight 1
!
queue 1
  meter-type packets
  pir 100
  type wrr
  weight 1
!
queue 2
  meter-type packets
  pir 600
  type wrr
  weight 2
--more--

```

Save running configuration to home directory

You can save the show running-configuration output to a file in the home directory folder by specifying a filename. A directory path in the home directory is not supported.

```

sonic# show running-configuration | save /home/admin/runningConfig.txt
%Error: cannot create regular file /home/admin//home/admin/runningConfig.txt: No such
file or Directory

sonic# show running-configuration | save runningConfig.txt
sonic# dir home:/
-----
Date(Last Modified)      Size(Bytes)      Type      Filename
-----
2023-08-30 04:18          32475           -         runningConfig.txt

```

View running configuration of Ethernet interface

```

sonic# show running-configuration interface Eth1/1
!
interface Eth1/1
mtu 9100
speed 100000
shutdown

```

View running BGP configuration

```

sonic# show running-configuration bgp
!
router bgp 1 vrf Vrf1
router-id 1.0.1.1
timers 60 180
!
address-family ipv4 unicast
  redistribute connected
  maximum-paths 16
  maximum-paths ibgp 1
!
address-family ipv6 unicast
  redistribute connected
  maximum-paths 1
  maximum-paths ibgp 1
!
address-family l2vpn evpn
  advertise ipv4 unicast
  advertise ipv6 unicast

```

```
rd 100:1
route-target both 101:1
```

Show configuration

To display the configured settings of an interface or Enterprise SONiC feature from a configuration mode in the CLI hierarchy, use the `show configuration` command.

Display Ethernet interface configuration

```
sonic# interface Eth 1/1
sonic(conf-if-Eth1/1)# show configuration
!
interface Eth1/1
  mtu 9100
  speed 100000
  switchport access Vlan 51
  switchport trunk allowed Vlan 52-56,58,60,62-64
```

Display port-channel configuration

```
sonic(conf)# interface PortChannel 5
sonic(conf-if-po5)# show configuration
!
interface PortChannel 5
  mtu 5000
  ip vrf forwarding VrfYellow
  ip address 6.6.6.1/24
```

Display BGP configuration

```
sonic(config)# router bgp 50 vrf VrfYellow
sonic(conf-router-bgp)# show configuration
!
router bgp 50 vrf VrfYellow
timers 500 3000
!
address-family ipv4 unicast
!
neighbor interface Eth1/2
  description bgp-neighbor
  capability dynamic passive
```

Display BGP neighbor configuration

```
sonic(conf-router-bgp)# neighbor interface Eth1/2
sonic(conf-router-bgp-neighbor)# show configuration
!
neighbor interface Eth1/2
description bgp-neighbor
capability dynamic passive
```

View software version

To display the software version and docker containers in the running Enterprise SONiC image, use the `show version` command.

```
sonic# show version

Software Version    : 4.2.0-Enterprise_Standard_Build86
Product            : Enterprise SONiC Distribution by Dell Technologies
Distribution       : Debian 10.13
Kernel             : 5.10.0-21-amd64
Config DB Version  : version 4_2_1
Build Commit       : 74b2bc3234
```

```

Build Date      : Mon Aug 21 13:23:13 UTC 2023
Built By       : sonicbld@bld-lvn-csg-05
Platform        : x86_64-dellemc_s5232f_c3538-r0
HwSKU          : DellEMC-S5232f-C32
ASIC           : broadcom
Hardware Version : A03
Serial Number   : CN0RC7V6CES009AQ0024
Uptime          : 09:00:16 up 16:11, 1 user, load average: 2.02, 1.93, 1.87
Mfg            : Dell EMC

```

REPOSITORY	TAG	IMAGE ID	SIZE
docker-database	4.2.0-Enterprise_Standard_Build86	3ebe9aa10b42	419MB
docker-database	latest	3ebe9aa10b42	419MB
docker-dhcp-relay-ent-base	4.2.0-Enterprise_Standard_Build86	82ffbee88c46	497MB
docker-dhcp-relay-ent-base	latest	82ffbee88c46	497MB
docker-eventd	4.2.0-Enterprise_Standard_Build86	ff2e09f72a3e	414MB
docker-eventd	latest	ff2e09f72a3e	414MB
docker-fpm-frr	4.2.0-Enterprise_Standard_Build86	12647d0ca6d7	520MB
docker-fpm-frr	latest	12647d0ca6d7	520MB
docker-gbsyncd-brcm	4.2.0-Enterprise_Standard_Build86	8daab22255f6	866MB
docker-gbsyncd-brcm	latest	8daab22255f6	866MB
docker-iccpd	4.2.0-Enterprise_Standard_Build86	b7e905a1aae7	496MB
docker-iccpd	latest	b7e905a1aae7	496MB
docker-l2mcd	4.2.0-Enterprise_Standard_Build86	720b78f29f61	493MB
docker-l2mcd	latest	720b78f29f61	493MB
docker-lldp	4.2.0-Enterprise_Standard_Build86	ef859e19a161	498MB
docker-lldp	latest	ef859e19a161	498MB
docker-nat	4.2.0-Enterprise_Standard_Build86	8ac8c9d30f23	495MB
docker-nat	latest	8ac8c9d30f23	495MB
docker-platform-monitor	4.2.0-Enterprise_Standard_Build86	b35ce8a887a9	620MB
docker-platform-monitor	latest	b35ce8a887a9	620MB
docker-router-advertiser	4.2.0-Enterprise_Standard_Build86	f92dc1c27271	413MB
docker-router-advertiser	latest	f92dc1c27271	413MB
docker-sflow	4.2.0-Enterprise_Standard_Build86	8be26528bbfc	495MB
docker-sflow	latest	8be26528bbfc	495MB
docker-snmp	4.2.0-Enterprise_Standard_Build86	b7c1043d7364	436MB
docker-snmp	latest	b7c1043d7364	436MB
docker-sonic-mgmt-framework	4.2.0-Enterprise_Standard_Build86	b194091ef381	659MB
docker-sonic-mgmt-framework	latest	b194091ef381	659MB
docker-sonic-telemetry	4.2.0-Enterprise_Standard_Build86	8f0f77e9eb16	597MB
docker-sonic-telemetry	latest	8f0f77e9eb16	597MB
docker-stp	4.2.0-Enterprise_Standard_Build86	e500d3348609	496MB
docker-stp	latest	e500d3348609	496MB
docker-swss-brcm-ent-base	4.2.0-Enterprise_Standard_Build86	47f5dc574e97	489MB
docker-swss-brcm-ent-base	latest	47f5dc574e97	489MB
docker-syncd-brcm-ent-base	4.2.0-Enterprise_Standard_Build86	dea461b0979e	833MB
docker-syncd-brcm-ent-base	latest	dea461b0979e	833MB
docker-tam	4.2.0-Enterprise_Standard_Build86	ba7cd7f7150e	499MB
docker-tam	latest	ba7cd7f7150e	499MB
docker-teamd	4.2.0-Enterprise_Standard_Build86	4904e698bdf3	493MB
docker-teamd	latest	4904e698bdf3	493MB
docker-udld	4.2.0-Enterprise_Standard_Build86	b6c62c9f9df4	499MB
docker-udld	latest	b6c62c9f9df4	499MB
docker-vrrp	4.2.0-Enterprise_Standard_Build86	9236602faf94	501MB
docker-vrrp	latest	9236602faf94	501MB

You can save the show version output to a file in the home directory folder by specifying a filename. A directory path in the home directory is not supported.

```

sonic# show version | save /home/admin/showVersion.txt
%Error: cannot create regular file /home/admin//home/admin/showVersion.txt: No such file or Directory

sonic# show version | save showVersion.txt
sonic# dir home:/
-----
Date(Last Modified)      Size(Bytes)    Type     Filename
-----
2023-08-30 04:22          6138          -        showVersion.txt
sonic# show version
! =====
! Started saving show command output at 30/08, 2023, 04:22:48 for command:

```

```

! show version | save showVersion.txt
! =====

Software Version   : 4.2.0-Enterprise_Premium_Build88
Product          : Enterprise SONiC Distribution by Dell Technologies
Distribution     : Debian 10.13
Kernel           : 5.10.0-21-amd64
Config DB Version: version_4_2_1
Build Commit     : a4854f952e
Build Date       : Tue Aug 22 13:15:42 UTC 2023
Built By         : sonicbld@bld-lvn-csg-03
Platform          : x86_64-dell EMC_s5296f_c3538-r0
HwSKU            : DellEMC-S5296f-P-25G-DPB
ASIC              : broadcom
Hardware Version : X01
Serial Number    : CN0WRHD6CES0085P0006
Uptime            : 04:22:48 up 9 min, 1 user, load average: 2.18, 3.44, 2.20
Mfg               : Dell EMC

REPOSITORY          TAG                IMAGE ID        SIZE
docker-database      latest             3ebe9aa10b42  419MB
docker-database      4.2.0-Enterprise_Premium_Build88 3ebe9aa10b42  419MB
docker-dhcp-relay-ent-advanced latest             d8eb10695f2b  497MB
docker-dhcp-relay-ent-advanced latest             d8eb10695f2b  497MB
docker-eventd         latest             ff2e09f72a3e  414MB
docker-eventd         4.2.0-Enterprise_Premium_Build88 ff2e09f72a3e  414MB
docker-fpm-frr        latest             04dd4b71414b  520MB
docker-fpm-frr        latest             04dd4b71414b  520MB
docker-gbsyncd-brcm   latest             a8df4e624fbf  866MB
docker-gbsyncd-brcm   4.2.0-Enterprise_Premium_Build88 a8df4e624fbf  866MB
docker-iccpd          latest             90d30107d553  496MB
docker-iccpd          4.2.0-Enterprise_Premium_Build88 90d30107d553  496MB
docker-l2mcd          latest             2ad55f762730  493MB
docker-l2mcd          4.2.0-Enterprise_Premium_Build88 2ad55f762730  493MB
docker-lldp            latest             b3becd72cc1b  498MB
docker-lldp            4.2.0-Enterprise_Premium_Build88 b3becd72cc1b  498MB
docker-nat             latest             7f0cb83634a7  495MB
docker-nat             4.2.0-Enterprise_Premium_Build88 7f0cb83634a7  495MB
docker-platform-monitor latest             b35ce8a887a9  620MB
docker-platform-monitor latest             b35ce8a887a9  620MB
docker-router-advertiser latest             f92dc1c27271  413MB
docker-router-advertiser latest             f92dc1c27271  413MB
docker-sflow            latest             7b9ca037c532  495MB
docker-sflow            4.2.0-Enterprise_Premium_Build88 7b9ca037c532  495MB
docker-snmp             latest             b7c1043d7364  436MB
docker-snmp             4.2.0-Enterprise_Premium_Build88 b7c1043d7364  436MB
docker-sonic-mgmt-framework latest             b194091ef381  659MB
docker-sonic-mgmt-framework latest             b194091ef381  659MB
docker-sonic-telemetry   latest             8f0f77e9eb16  597MB
docker-sonic-telemetry   latest             8f0f77e9eb16  597MB
docker-stp              latest             3077b23eec28  496MB
docker-stp              4.2.0-Enterprise_Premium_Build88 3077b23eec28  496MB
docker-swss-brcm-ent-advanced latest             bef73c559f0a  489MB
docker-swss-brcm-ent-advanced latest             bef73c559f0a  489MB
docker-syncd-brcm-ent-advanced latest             52d2189d8c96  836MB
docker-syncd-brcm-ent-advanced latest             52d2189d8c96  836MB
docker-tam              latest             ba7cd7f7150e  499MB
docker-tam              4.2.0-Enterprise_Premium_Build88 ba7cd7f7150e  499MB
docker-teamd             latest             076a9c68e0b6  493MB
docker-teamd             latest             076a9c68e0b6  493MB
docker-udld              latest             879aa8a5b74c  499MB
docker-udld              latest             879aa8a5b74c  499MB
docker-vrrp              latest             31b9df5070ae  501MB
docker-vrrp              latest             31b9df5070ae  501MB

```

View system hardware

To monitor switch hardware operation, you can display information about the system status.

View system information

```
sonic# show system
-----
Attribute          Value/State
-----
Boot Time          :00:12:56
CurrentDatetime   :2023-03-17T21:25:24Z
Hostname           :sonic
```

View CPU usage

```
sonic# show system cpu
-----
CPU                %KERNEL    %USER      %IDLE
-----
CPU-ALL            3          9          87
CPU-0              2          8          88
CPU-1              2          10         86
CPU-2              3          9          87
CPU-3              3          10         86
```

View memory usage

```
sonic# show system memory
-----
Attribute          Value/State
-----
Used               :1304976
Total              :8162872
```

View system processes

Use the `show system processes [pid number]` command to display the ID of processes running on the system and information about a specific process. Use the `show system processes [cpu | mem-usage | mem-util]` options to display process information that is sorted by CPU or memory usage.

```
sonic# show system processes
-----
PID    %CPU    %MEMORY   MEM-USAGE (Bytes)  NAME
-----
1      0        0          59203584        /sbin/init
2      0        0          0                  [kthreadd]
3      0        0          0                  [ksoftirqd/0]
5      0        0          0                  [kworker/0:0H]
7      0        0          0                  [rcu_sched]
8      0        0          0                  [rcu_bh]
9      0        0          0                  [migration/0]
10     0        0          0                  [lru-add-drain]
11     0        0          0                  [watchdog/0]
12     0        0          0                  [cpuhp/0]
13     0        0          0                  [cpuhp/1]
14     0        0          0                  [watchdog/1]
15     0        0          0                  [migration/1]
16     0        0          0                  [ksoftirqd/1]
18     0        0          0                  [kworker/1:0H]
19     0        0          0                  [cpuhp/2]
20     0        0          0                  [watchdog/2]
21     0        0          0                  [migration/2]
22     0        0          0                  [ksoftirqd/2]
24     0        0          0                  [kworker/2:0H]
25     0        0          0                  [cpuhp/3]
```

```
sonic# show system processes pid 1191
-----
```

Attribute	Value/State
Cpu Usage System	:720
Cpu Usage User	:574
Cpu Utilization	:1
Memory Usage	:11726848
Memory Utilization	:0
Name	:/sbin/init
Pid	:1
Start Time	:2023-03-17 21:12:28+0000
Uptime	:00:19:12

```
sonic# show system processes mem-util
```

PID	%CPU	%MEMORY	MEM-USAGE (Bytes)	NAME
4439	16	5	1832013824	/usr/bin/synccd
26510	0	3	1534111744	/usr/sbin/rest_server
6456	0	3	917860352	telemetry
728	0	1	985980928	/usr/bin/dockerd
1	0	0	59203584	/sbin/init
10	0	0	0	[lru-add-drain]
10019	0	0	12062720	/bin/bash
10023	0	0	552636416	docker
10042	0	0	59576320	/bin/bash
10053	0	0	306921472	/usr/sbin/cli/clis
101	0	0	0	[ata_sff]
1028	0	0	111722496	containerd-shim
1044	0	0	59650048	/usr/bin/python
11	0	0	0	[watchdog/0]
1140	0	0	42156032	python
1141	0	0	256139264	/usr/sbin/rsyslogd
1142	0	0	61546496	/usr/bin/redis-server
1143	5	0	95100928	/usr/bin/redis-server
1144	0	0	65740800	/usr/bin/redis-server
1145	1	0	69935104	/usr/bin/redis-server
1191	0	0	12165120	/bin/bash

View EEPROM information

```
sonic# show platform syseeprom
```

Attribute	Value/State
Platform	:x86_64-dell_z9246f_c2538-r0
Hardware Version	:A02-
Product Name	:Z9264F-ON
Location	:Slot 1
Mfg Date	:2017-08-07
Mfg Name	:CES00
Base Mac Address	:34:17:EB:2C:D7:00
Diag Version	:3.23.4.1
Mac Addresses	:384
Manufacture Country	:CN
Onie Version	:3.23.1.0-7
Service Tag	:847RG02
Vendor Name	:DELL
Part Number	:OKY5C4
Serial Number	:CN0KY5C4CES007770009

View fan, temperature, power supply, and adapter status

```
sonic# show platform environment
```

```
Onboard Temperature Sensors:
  Broadcom Temp:          38 degrees C
  CPU Temp:                33 degrees C
  Inlet Airflow Temp:      27 degrees C
  PT_Left_temp:            26 degrees C
  PT_Mid_temp:              31 degrees C
  PT_Right_temp:            31 degrees C
```

```

Fan Trays:
Fan Tray 1:
    Airflow:                    Exhaust
    Fan1 Speed:                 8520 RPM
    Fan1 State:                 Normal
    Fan2 Speed:                 7680 RPM
    Fan2 State:                 Normal
Fan Tray 2:
    Airflow:                    Exhaust
    Fan1 Speed:                 8400 RPM
    Fan1 State:                 Normal
    Fan2 Speed:                 7680 RPM
    Fan2 State:                 Normal
Fan Tray 3:
    Airflow:                    Exhaust
    Fan1 Speed:                 8640 RPM
    Fan1 State:                 Normal
    Fan2 Speed:                 7920 RPM
    Fan2 State:                 Normal
Fan Tray 4:
    Airflow:                    Exhaust
    Fan1 Speed:                 8640 RPM
    Fan1 State:                 Normal
    Fan2 Speed:                 7800 RPM
    Fan2 State:                 Normal

```

```

PSUs:
PSU1:
    Airflow:                    Exhaust
    FAN AirFlow Temperature:    28 degrees C
    FAN Normal Temperature:    36 degrees C
    FAN RPM:                   7680 RPM
    Input Current:              0.32 Amps
    Input Power:                80 Watts
    Input Voltage:              209 Volts
    Output Current:             5.50 Amps
    Output Power:               70 Watts
    Output Voltage:             12.40 Volts
PSU2:
    Airflow:                    Exhaust
    FAN AirFlow Temperature:    28 degrees C
    FAN Normal Temperature:    45 degrees C
    FAN RPM:                   7440 RPM
    Input Current:              0.32 Amps
    Input Power:                70 Watts
    Input Voltage:              209 Volts
    Output Current:             4.50 Amps
    Output Power:               50 Watts
    Output Voltage:             12.50 Volts

```

View fan status

In show platform fanstatus output:

- OK — Fan is installed and operation.
- NOT OK — Fan is installed and not operational.
- NOT PRESENT — Fan is not installed.

```

sonic# show platform fanstatus
-----
Fan          Status      Speed (RPM)   Direction
-----
FAN 1        OK          7214          exhaust
FAN 2        OK          7419          exhaust
FAN 3        OK          7136          exhaust
FAN 4        OK          7447          exhaust
FAN 5        OK          7295          exhaust
FAN 6        OK          7547          exhaust
FAN 7        OK          7377          exhaust
FAN 8        OK          7461          exhaust

```

FAN 9	OK	7336	exhaust
FAN 10	OK	7561	exhaust

View temperature

```
sonic# show platform temperature
TH - Threshold
-----
Name      Temperature  High   Low    Critical  Critical  Warning  Timestamp
          TH       TH     High TH   Low TH
-----
ASIC On-board...  41.2    N/A    N/A    N/A      N/A      false   20200917 18:49:57
ASIC On-board Rear 41.1    85     0      N/A      N/A      false   20200917 18:52:57
CPU Core 0        35      98     0      N/A      N/A      false   20200917 18:52:57
CPU Core 1        35      98     0      N/A      N/A      false   20200917 18:52:57
CPU Core 2        39      98     0      N/A      N/A      false   20200917 18:52:57
CPU Core 3        38      98     0      N/A      N/A      false   20200917 18:52:57
CPU On-board      36.8    100    0      N/A      N/A      false   20200917 18:52:57
System Front Left 24.8    50     0      N/A      N/A      false   20200917 18:52:57
System Front Right 25.2    50     0      N/A      N/A      false   20200917 18:52:57
```

Use the `show platform temperature detail` command to display the name of a temperature sensor that is not fully displayed; for example:

```
sonic# show platform temperature detail

Platform Temperature Sensor Details
-----
Sensor name:           ASIC On-board Back-panel port recv sensor
Temperature:          41.2
High threshold:       N/A
Low threshold:        N/A
Critical High threshold: N/A
Critical Low threshold: N/A
Warning status:       False
Timestamp:            2020-09-17T18:49:57Z

Sensor name:           ASIC On-board Rear
Temperature:          41.1
High threshold:       85
Low threshold:        0
Critical High threshold: N/A
Critical Low threshold: N/A
Warning status:       False
Timestamp:            2020-09-17T18:52:57Z
...
```

View power supply status

In `show platform psustatus` output:

- OK — Power supply is installed and operation.
- NOT OK — Power supply is installed and not operational.
- NOT PRESENT — Power supply is not installed.

```
sonic# show platform psustatus
-----
PSU                  Status
-----
PSU 1                OK
PSU 2                NOT OK
PSU 3                NOT PRESENT
```

The `show platform psusummary` command displays N/A for Output Current, Output Power, Output Voltage on the N3248PXE-ON and E3248PXE-ON platforms for external PSUs.

```
sonic# show platform psusummary
PSU 1:
  Description      :DPS-1600AB-34 C
  Fans             :1
```

```

Mfg Name :DELTA
Name :PSU 1
Oper Status :OK
Serial Number :xxxxxxxxxxxx
Status LED: :None
Type (AC/DC) :AC
Input Current (A) :3.00
Input Power (W) :669.00
Input Voltage (V) :225.80
Output Current (A) :11.30
Output Power (W) :629.00
Output Voltage (V) :55.60
Fan Speed (RPM) :8352
Fan Direction :exhaust
Temperature :35.40
PSU 2:
Description :DPS-1600AB-34 C
Fans :1
Mfg Name :DELTA
Name :PSU 2
Oper Status :OK
Serial Number :xxxxxxxxxxxx
Status LED: :None
Type (AC/DC) :AC
Input Current (A) :3.00
Input Power (W) :681.00
Input Voltage (V) :225.20
Output Current (A) :11.70
Output Power (W) :646.00
Output Voltage (V) :55.50
Fan Speed (RPM) :7920
Fan Direction :exhaust
Temperature :33.20
PSU 3:
Description :04JR64
Fans :0
Mfg Name :DELTA
Name :PSU 3
Oper Status :OK
Serial Number :xxxxxxxxxxxxxx
Status LED: :None
Type (AC/DC) :Unknown
Input Current (A) :N/A
Input Power (W) :N/A
Input Voltage (V) :N/A
Output Current (A) :22.90
Output Power (W) :1276.00
Output Voltage (V) :55.70
Fan Speed (RPM) :0
Fan Direction :none
Temperature :N/A

```

View solid-state drive health

An Enterprise SONiC switch uses a solid-state drive (SSD) for event logging and storage. Use the `show platform ssdhealth` command to display the health percentage of the drive. A low health percentage in the output indicates that the SSD is nearing end-of-life and should be replaced to avoid a switch crash.

```

sonic# show platform ssdhealth
Firmware Version : SBR13067
Health : 100.0
Device Model : SFSA120GM3AA2TO-C-OC-23P-DEL
Serial Number : 000060203793A4000120
Temperature : 32C

```

View core service status

To view the status of core system services, enter the SONiC CLI command `show system status`.

sonic# show system status brief	System is ready	Service-Status	App-Ready-Status	Down-Reason	
sonic# show system status	System is ready	Service-Name	Service-Status	App-Ready-Status	Down-Reason
		swss	OK	OK	-
		bgp	OK	OK	-
		teamd	OK	OK	-
		pmon	OK	OK	-
		syncd	OK	OK	-
		database	OK	OK	-
		gbsyncd	OK	OK	-
		caclmgrd	OK	OK	-
		ccd	OK	OK	-
		config-chassisdb	OK	OK	-
		config-setup	OK	OK	-
		containerd	OK	OK	-
		critical-monitoring	OK	OK	-
		cron	OK	OK	-
		database-chassis	OK	OK	-
		db-post-startup	OK	OK	-
		determine-reboot-cause	OK	OK	-
		dhcp_relay	OK	OK	-
		disk-log-rotate-daemon	OK	OK	-
		docker	OK	OK	-
		eventd	OK	OK	-
		hamd	OK	OK	-
		histogram	OK	OK	-
		hostcfgd	OK	OK	-
		hostname-config	OK	OK	-
		iccpd	OK	OK	-
		in-memory-log-rotate-daemon	OK	OK	-
		in-memory	OK	OK	-
		interfaces-config	OK	OK	-
		kdump-tools	OK	OK	-
		l2mcd	OK	OK	-
		lacp_helper	OK	OK	-
		lldp	OK	OK	-
		macsec	OK	OK	-
		mux-ctrl	OK	OK	-
		netfilter-persistent	OK	OK	-
		ntp-config	OK	OK	-
		ntp	OK	OK	-
		opennsl-modules	OK	OK	-
		platform-init	OK	OK	-
		platform-modules-e3248p	OK	OK	-
		portinitdone	OK	OK	-
		procdockerstatsd	OK	OK	-
		radv	OK	OK	-
		ras-mc-ctl	OK	OK	-
		resrcmgrd	OK	OK	-
		rsyslog-config	OK	OK	-
		rsyslog	OK	OK	-
		sflow	OK	OK	-
		snmp	OK	OK	-
		sonic-hostservice	OK	OK	-
		ssh	OK	OK	-
		stp	OK	OK	-
		sysmonitor	OK	OK	-
		system-health	OK	OK	-
		tpcm@default	OK	OK	-
		udld	OK	OK	-
		updategraph	OK	OK	-
		vrrp	OK	OK	-
		warmboot-finalizer	OK	OK	-

```

watchdog-control          OK
ztp-config                OK
mgmt-framework           OK
telemetry                 OK
sonic#

```

Service-Name	Service-Status	App-Ready-Status	Down-Reason
swss	OK	OK	-
2023-06-30 04:57:11			
bgp	OK	OK	-
2023-06-30 04:57:18			
teamd	OK	OK	-
2023-06-30 04:55:45			
pmon	OK	OK	-
2023-06-30 04:56:57			
syncthreads	OK	OK	-
2023-06-30 04:56:11			
database	OK	OK	-
mgmt-framework	OK	OK	-
2023-06-30 04:55:34			
gbsyncd	OK	OK	-
cac1mgard	OK	OK	-
ccd	OK	OK	-
config-chassisdb	OK	OK	-
config-setup	OK	OK	-
containerd	OK	OK	-
critical-monitoring	OK	OK	-
cron	OK	OK	-
database-chassis	OK	OK	-
db-post-startup	OK	OK	-
determine-reboot-cause	OK	OK	-
dhcp_relay	OK	OK	-
2023-06-30 04:57:04			
disk-log-rotate-daemon	OK	OK	-
docker	OK	OK	-
eventd	OK	OK	-
2023-06-30 04:55:04			
hamd	OK	OK	-
histogram	OK	OK	-
hostcfgd	OK	OK	-
hostname-config	OK	OK	-
iccpd	OK	OK	-
2023-06-30 04:56:30			
in-memory-log-rotate-daemon	OK	OK	-
in-memory	OK	OK	-
interfaces-config	OK	OK	-
kdump-tools	OK	OK	-
l2mcd	OK	OK	-
2023-06-30 04:56:57			
lacp_helper	OK	OK	-
lldp	OK	OK	-
2023-06-30 04:56:30			
nat	OK	OK	-
2023-06-30 04:56:28			
netfilter-persistent	OK	OK	-
ntp-config	OK	OK	-
ntp	OK	OK	-
opennsl-modules	OK	OK	-
platform-init	OK	OK	-
platform-modules-s5248f	OK	OK	-
portinitdone	OK	OK	-
proc docker statsd	OK	OK	-
radv	OK	OK	-
2023-06-30 04:57:03			
ras-mc-ctl	OK	OK	-
resrcmgrd	OK	OK	-
rsyslog-config	OK	OK	-
rsyslog	OK	OK	-
sflow	OK	OK	-

```

2023-06-30 04:57:02
snmp                                OK      OK      -
2023-06-30 04:57:45
sonic-hostservice                     OK      OK      -
ssh                                   OK      OK      -
stp                                   OK      OK      -
2023-06-30 04:56:30
sysmonitor                           OK      OK      -
tam                                    OK      OK      -
2023-06-30 04:57:28
telemetry                            OK      OK      -
2023-06-30 04:56:25
tpcm@default                          OK      OK      -
udld                                  OK      OK      -
2023-06-30 04:56:30
updategraph                           OK      OK      -
vrrp                                  OK      OK      -
2023-06-30 04:57:01
warmboot-finalizer                    OK      OK      -
watchdog-control                      OK      OK      -
ztp-config                            OK      OK      -
tpcm@mgmt                             OK      OK      -
sonic#

```

If a core service is down, log in to the Linux shell and run the following SONiC CLI command to get more information.

```

admin@sonic:~$ sudo systemctl status pmon
● pmon.service - Platform monitor container
  Loaded: loaded (/usr/lib/systemd/system/pmon.service; enabled; vendor preset:
  Active: inactive (dead) since Thu 2021-02-25 14:35:46 UTC; 5s ago
    Process: 14792 ExecStop=/usr/bin/pmon.sh stop (code=exited, status=0/SUCCESS)
    Process: 14575 ExecStart=/usr/bin/pmon.sh wait (code=exited, status=0/SUCCESS)
    Process: 14298 ExecStartPre=/usr/bin/pmon.sh start (code=exited, status=0/SUCCESS)
   Main PID: 14575 (code=exited, status=0/SUCCESS)
     CPU: 3.857s

```

To recover a core service that is down, enter the `sudo systemctl restart core-service` command.

```

admin@sonic:~$ sudo systemctl restart core-service

```

To verify that the core service is reactivated, use the `sudo systemctl status pmon` or `show system status` command.

```

admin@sonic:~$ sudo systemctl status pmon
● pmon.service - Platform monitor container
  Loaded: loaded (/usr/lib/systemd/system/pmon.service; enabled; vendor preset:
  Active: activating (start-pre) since Thu 2021-02-25 14:40:39 UTC; 2s ago
    Process: 18061 ExecStop=/usr/bin/pmon.sh stop (code=exited, status=0/SUCCESS)
    Process: 17399 ExecStart=/usr/bin/pmon.sh wait (code=exited, status=0/SUCCESS)
   Main PID: 17399 (code=exited, status=0/SUCCESS); Control PID: 18180 (pmon.sh)
      Tasks: 8 (limit: 4915)
     Memory: 52.0M
        CPU: 2.133s
       CGroup: /system.slice/pmon.service
                   └─control
                         ├─18180 /bin/bash /usr/bin/pmon.sh start
                         ├─18223 sudo /usr/bin/sysupdate.py

```

If the core service continues to be down, contact Dell technical support.

View system firmware

To verify the firmware installed on a switch, enter the `show platform firmware` command.

View system firmware

```
sonic# show platform firmware
-----
-
Chassis   Module   Component      Version      Description
-----
-
Z9432F-ON N/A     BIOS          3.51.0.D-6    Performs initialization of hardware...
                                BMC          3.03         Platform management controller for...
                                FPGA         2.0          Used for managing the system LEDs
                                Secondary CPLD 1 1.2          Used for managing SFP28/QSFP28 port...
                                Secondary CPLD 2 1.2          Used for managing SFP28/QSFP28 port...
                                System CPLD    1.2          Used for managing the CPU power
                                                sequence...
```

In the `show platform firmware` output, firmware components are displayed in alphabetical order; for example, BIOS through FPGA. The alphabetical order corresponds to the numbers 1 to 5, in which the first component — in this example, BIOS — is number 1, the second component — BMC — is number 2, and so on. Enter the number that corresponds to an alphabetically ordered component to retrieve information about the component in a REST API call - see [REST API examples](#).

```
sonic# show platform firmware detail
-----
Platform Firmware Information
-----
Chassis:           Z9432F-ON
Module:            N/A
Component:         BIOS
Firmware Version: 3.51.0.D-6
Description:        Performs initialization of hardware components during booting

Chassis:           Z9432F-ON
Module:            N/A
Component:         BMC
Firmware Version: 3.03
Description:        Platform management controller for on-board temperature
                    monitoring, in-chassis power, Fan and LED control

Chassis:           Z9432F-ON
Module:            N/A
Component:         FPGA
Firmware Version: 2.0
Description:        Used for managing the system LEDs

Chassis:           Z9432F-ON
Module:            N/A
Component:         Secondary CPLD 1
Firmware Version: 1.2
Description:        Used for managing SFP28/QSFP28 port transceivers (SFP28 1-24,
                    QSFP28 1-4)

Chassis:           Z9432F-ON
Module:            N/A
Component:         Secondary CPLD 2
Firmware Version: 1.2
Description:        Used for managing SFP28/QSFP28 port transceivers (SFP28 25-48,
                    QSFP28 5-8)

Chassis:           Z9432F-ON
Module:            N/A
Component:         System CPLD
Firmware Version: 1.2
Description:        Used for managing the CPU power sequence and CPU states
```

Restore factory defaults

To restore the system factory defaults, you can roll back the switch configuration to the factory default configuration by using the `write erase` command. To complete the configuration erase operation, reboot the switch using the `reboot` command.

```
sonic# write erase [boot | install]
```

- `write erase` — Deletes the startup configuration JSON file `/etc/sonic/config_db.json` and all application configuration files in the `/etc/sonic` directory. The management interface configuration in the startup configuration file is retained after the switch reboots to allow access to the switch using the management IP address. The switch reloads with a factory-default configuration — see [Default startup configuration](#).
- `write erase boot` — Deletes the startup configuration JSON file, all application configuration files in the `/etc/sonic` directory, and the Management interface configuration in the startup configuration JSON file. The switch reloads with a factory-default configuration.
- `write erase install` — Deletes the startup configuration JSON file, all application configuration files in the `/etc/sonic` directory, all configuration changes made in the running configuration, and all user-installed packages. The switch reverts to the state of a newly installed image. After you reload the switch, if ZTP is enabled, the switch starts DHCP discovery to locate and download a switch configuration file — see [Zero touch provisioning](#).
- `no write erase` — A `write erase` operation requires a system reload in order to take effect. Before you reload the switch, you can cancel the configuration removal by entering the `no write erase` command.

Reset Enterprise SONiC password

If you lose or forget an Enterprise SONiC username password, including the `admin` password, you can reset the password by the following procedure.

1. Connect to the serial or USB console port. The serial settings are 115200 baud, 8 data bits, and no parity.
2. Reboot or power ON the switch. If you reboot, be sure to save any unsaved configuration changes by entering the `write memory` command in Exec mode.
3. Wait for the GRUB menu to appear.

```
GNU GRUB version 2.02
+-----+
| *SONiC-OS-4.0.5-Enterprise_Base |
| ONIE |
| |
| |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
```

4. In the GRUB menu, select the Enterprise SONiC version. Then press `e` to open the GRUB editor. Use the arrow keys to navigate to the line that starts with `linux /image-`. This line becomes highlighted.

```
GNU GRUB version 2.02
/-----
|setparams 'SONiC-OS-4.0.5-Enterprise_Base'
|
| search --no-floppy --label --set=root SONiC-OS
| echo 'Loading SONiC-OS OS kernel ...'
| insmod gzio
| if [ x = xxen ]; then insmod xzio; insmod lzopio; fi
| insmod part_msdos
| insmod ext2
| linux /image-4.0.5-Enterprise_Base/boot/vmlinuz-4.19.0-9-2-amd64\
| root=UUID=1ef6ad59-e7cf-4db4-94db-68fe15d8a089 rw console=tty0 console=tty\
| S0,115200n8 quiet intel_idle.max_cstate=0 net.ifnames=0 biosdevname=0\
| loop=image-4.0.5-Enterprise_Base/fs.squashfs loopfstype=squashfs\
| crashkernel=0M-2G:2\
\-----
Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
command-line or ESC to discard edits and return to the GRUB menu.
```

i **NOTE:** If the arrow keys do not work in GRUB editor, use the following key combinations to navigate to the line:

- To go up: Ctrl+P
- To go down: Ctrl+N
- To go left: Ctrl+B
- To go right: Ctrl+F

5. In the highlighted line that starts with `linux /image-`, type `init=/bin/bash` between `r/w` and `console=tty0` as shown here:

```
GNU GRUB version 2.02
-----
|setparams 'SONiC-OS-4.0.5-Enterprise_Base'
|
| search --no-floppy --label --set=root SONiC-OS
| echo 'Loading SONiC-OS OS kernel ...'
| insmod gzio
| if [ x = xxen ]; then insmod xzio; insmod lzopio; fi
| insmod part_msdos
| insmod ext2
| linux /image-4.0.5-Enterprise_Base/boot/vmlinuz-4.19.0-9-2-amd64\
|root=UUID=1ef6ad59-e7cf-4db4-94db-68fe15d8a089 rw init=/bin/bash console=tty0\
|console=ttyS0,115200n8 quiet intel_idle.max_cstate=0 net.ifnames=0\
|biosdevname=0 loop=image-4.0.5-Enterprise_Base/fs.squashfs\
|loopfstype=squashfs crashkernel=0M-2G:2\
\-----
Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
command-line or ESC to discard edits and return to the GRUB menu.
```

i **NOTE:** To undo any text you have typed and return to the GRUB menu without saving, press `Esc`.

This text insertion does not survive a reboot. You do not have to re-edit Grub after you recover your password and use it to log in again.

6. Reboot by pressing F10 or Ctrl+X. A sample boot sequence:

```
Booting a command listBooting a command list
Loading SONiC-OS OS kernel ...Loading SONiC-OS OS kernel ...
Loading SONiC-OS OS initial ramdisk ...Loading SONiC-OS OS initial ramdisk ...
AF,
DXE_EXIT_BOOT_SERVICES(03101019)
B, B, bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
root@(none) :/#
```

7. To reset the password, use the Linux command `passwd username`; for example:

```
root@(none) :/# passwd admin
Enter new UNIX password: xxxxxx
Retype new UNIX password: xxxxxx
passwd: password updated successfully
```

8. Reboot the switch by entering the Linux command `echo "b" > /proc/sysrq-trigger` or `reboot -f`; for example:

```
root@(none) :/# echo "b" > /proc/sysrq-trigger
```

Or

```
root@(none) :/# reboot -f
```

After the switch reboots, you can log in with your username and the new password.

System management and setup

Software image management	Installs or upgrades an Enterprise SONiC image and net-boot image (see Software image management).
Interface naming modes	To easily identify the front-panel port associated with an interface, change from the default native interface-naming mode to standard mode.
RBAC	Sets up controls for system access and user authorization based on defined roles (see Role-based access control).
AAA	Secures your network from unauthorized access using authentication, authorization, and accounting (AAA) services (see Authentication, authorization, and accounting).
Network Time Protocol	Synchronizes the system clock with NTP servers to receive accurate time updates (see Network time protocol).
DNS server	Translates host names to IP addresses (see DNS server).
Fast reboot	Reboots a switch quickly during maintenance or system upgrade to minimize the disruption of data transmission (see Fast reboot).
Simple Network Management Protocol	Monitors and manages switch performance (see Simple Network Management Protocol).
Dynamic Host Configuration Protocol	Simplifies assignment of IP addresses and other information to network devices (See Dynamic Host Configuration Protocol).
Chassis locator LED	Identifies a switch in a multi-switch deployment (see Chassis locator LED).

For information about how to set up a management network that is separate from your production network, see [Management Networks for Dell Networking](#).

Topics:

- Software image management
- Firmware installation
- Install a software patch
- Using USB storage media
- System file management
- Configure user login lockout
- Interface naming modes
- Role-based access control
- Authentication, authorization, and accounting
- Network Time Protocol
- DNS server
- Fast reboot
- Simple Network Management Protocol
- Dynamic Host Configuration Protocol
- Third party containers
- L2 and L3 switch profiles
- L2/L3 host and route scaling
- Cut-through switching
- Configure CPU polling interval

Software image management

SONiC allows you to install a maximum of two software images. The available images can be the current running image or the next-boot image. The current and next-boot images may be the same.

Use `show image list` to view the available images.

```
sonic# show image list
Current: SONiC-OS-4.2.0-Edge_Standard_Build241
Next: SONiC-OS-4.2.0-Edge_Standard_Build241
Available:
SONiC-OS-4.2.0-Edge_Standard
SONiC-OS-4.2.0-Edge_Standard_Build241
```

Install or upgrade system image

1. Use the `copy` command to copy an image file from a source location to the local file system or a network server.

```
sonic# copy sourcefilepath destinationfilepath
```

- `sourcefilepath` — Enter the source file path as the URL of a remote server using an FTP, HTTP or HTTPS, or SCP download.
 - `ftp://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]`
 - `http[s]://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]`
 - `scp://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]`
- `destination-path` — Enter the destination file path as a local directory (`home:filepath`) or a file path on an FTP, HTTP or HTTPS, or SCP/SSH server.

For example:

```
sonic# copy scp://userid:passwd@hostip/
tftpboot/Enterprise SONiC OS_4.2.0_Edge_Standard.bin home://
Enterprise SONiC OS_4.2.0_Edge_Standard.bin
```

2. Save the running configuration to the startup configuration before you start the image upgrade.

```
sonic# copy run start
```

3. Use the `image install` command to install the image on the switch. The installed image is stored as the next-boot image.

```
sonic# image install file-url
```

Enter the `file-url` location of the image in one of the following formats:

- `http[s]://hostip:/filepath` — Install the image from a remote HTTP or HTTPS server.
- `file://filepath` — Install the image from the local or a USB file system.

For example:

```
sonic# image install file://home/admin/Enterprise SONiC OS_4.2.0_Edge_Standard.bin
%Info: Check 'show image status' for image install progress.
```

(i) NOTE: If the current running image contains any modified text files or installed custom packages, they are not available in a different image. Back up the modified files and reinstall the packages after downloading a new image. Also, if you do not reboot the switch after using the `image install` command and continue to make configuration changes, these changes are lost when the switch reloads with the new image.

4. To view the image installation progress, use the `show image status` command. The command output should display `Global operation status : GLOBAL_STATE_SUCCESS`. The following global operation statuses may be displayed:
 - `GLOBAL_STATE_DOWNLOAD` during the image downloading process
 - `GLOBAL_STATE_INSTALL` during the image installation process

- GLOBAL_STATE_SUCCESS after a successful image installation.

```
sonic# show image status
-----
Global operation status : GLOBAL_STATE_DOWNLOAD
-----
File operation status : TRANSFER_STATE_SUCCESS
File size(bytes) : 1727633390
File transfer bytes : 1727633390
File progress : 100%
Transfer start time : 2021-10-28 09:22:33+0000
Transfer end time : 2021-10-28 09:27:57+0000

sonic# show image status
-----
Global operation status : GLOBAL_STATE_SUCCESS
-----
File operation status : TRANSFER_STATE_SUCCESS
File size(bytes) : 1727633390
File transfer bytes : 1727633390
File progress : 100%
Transfer start time : 2021-10-28 09:22:33+0000
Transfer end time : 2021-10-28 09:27:57+0000
-----
Install operation status : INSTALL_STATE_SUCCESS
Install start time : 2021-10-28 09:27:57+0000
Install end time : 2021-10-28 09:29:21+0000
```

- GLOBAL_STATE_FAILED if you incorrectly specified the image file or entered the wrong server IP address; for example:
- Load the newly installed, next-boot image by rebooting the switch.

```
sonic# reboot
reboot in process .....
Waiting for the reboot operation to complete
11, 32, 15, 00068001, 19, 00068000,
```

Examples: Image install

```
sonic# image install http://192.168.100.1/tftpboot/
Enterprise SONIC OS 4.2.0 Enterprise Premium.bin
%Info: Check 'show image status' for image install progress.

sonic# show image status
-----
Global operation status : GLOBAL_STATE_FAILED
-----
File operation status : TRANSFER_DOWNLOAD_FAILED(No response from remote machine)
File size(bytes) : 0
File transfer bytes : 0
File progress : 0%
Transfer start time : N/A
Transfer end time : N/A
```

```
sonic# image install http://192.168.100.1/tftpboot/
Enterprise SONIC OS 4.2.0 Enterprise Premium.bin
%Error: Image file not found on remote machine. Aborting...

sonic# show image status
-----
Global operation status : GLOBAL_STATE_FAILED
-----
File operation status : TRANSFER_DOWNLOAD_FAILED(Image file not present on remote
machine)
File size(bytes) : 0
File transfer bytes : 0
File progress : 0%
Transfer start time : N/A
Transfer end time : N/A
```

To interrupt and cancel an active image installation, use the `image install cancel` command.

```
sonic# image install http://192.168.100.1/tftpboot/
Enterprise SONiC OS 4.2.0 Enterprise Premium.bin
%Info: Check 'show image status' for image install progress.

sonic# show image status
-----
Global operation status : GLOBAL_STATE_DOWNLOAD
-----
File operation status : TRANSFER_DOWNLOAD
File size(bytes) : 1727633390
File transfer bytes : 288931840
File progress : 16%
Transfer start time : 2021-10-28 09:35:31+0000
Transfer end time : 1970-01-01 00:00:00+0000
-----
Install operation status : INSTALL_IDLE
Install start time : 1970-01-01 00:00:00+0000
Install end time : 1970-01-01 00:00:00+0000

sonic# image install cancel
Cancel the image install process, continue? [y/N]:y

sonic# show image status
-----
Global operation status : GLOBAL_STATE_IDLE
-----

sonic# show image list
Current: SONiC-OS-dell_sonic_share.344-ff0c897f4
Next: SONiC-OS-dell_sonic_share.344-ff0c897f4
Available:
SONiC-OS-dell_sonic_share.344-ff0c897f4
```

Set next-boot image

You can change the next-boot image by entering the image filename that is displayed in the `show image list` output. You can set the same image as the current and next-boot image. To load the next-boot image, reload the switch.

```
sonic# show image list
Current: SONiC-OS-4.1.0-Enterprise_Standard
Next: SONiC-OS-4.1.0-Enterprise_Standard
Available:
SONiC-OS-4.1.0-Enterprise_Standard
SONiC-OS-4.2.0-Enterprise_Standard
sonic# image set-default SONiC-OS-4.2.0-Enterprise_Standard
sonic# show image list
Current: SONiC-OS-4.1.0-Enterprise_Standard
Next: SONiC-OS-4.2.0-Enterprise_Standard
Available:
SONiC-OS-4.1.0-Enterprise_Standard
SONiC-OS-4.2.0-Enterprise_Standard
```

Reload system image

To reboot the switch and load the next-boot image:

```
sonic# reboot

System configuration has been modified. Save? [yes/no]:yes
Saving system configuration

Proceed to reboot the system? [confirm yes/no]:yes
```

Remove an image

You can delete an unused SONiC image by entering the image filename that is displayed in the show image list output. You cannot remove the current running image.

```
sonic# show image list
Current: SONiC-OS-dell_sonic_share.982-7726cbde5
Next: SONiC-OS-dell_sonic_share.982-7726cbde5
Available:
SONiC-OS-dell_sonic_4.x_share.54-a7ec3a969
SONiC-OS-dell_sonic_share.982-7726cbde5

sonic# image remove SONiC-OS-dell_sonic_4.x_share.54-a7ec3a969
Remove image SONiC-OS-dell_sonic_4.x_share.54-a7ec3a969? [y/N]:y

sonic# show image list
Current: SONiC-OS-dell_sonic_share.982-7726cbde5
Next: SONiC-OS-dell_sonic_share.982-7726cbde5
Available:
SONiC-OS-dell_sonic_share.982-7726cbde5
```

Firmware installation

The ONIE loader allows you to install different firmware available in a platform, such as ONIE, BIOS, CPLD, BMC, and so on. Enterprise SONiC allows you to stage firmware packages using the MF-CLI.

You can install a firmware package from the ONIE prompt or from within Enterprise SONiC. This section describes the procedure to install a firmware package using the MF-CLI.

Important information for installing firmware

- Although you use an Enterprise SONiC command to stage a firmware file, ONIE installs the firmware package after you reload the switch.
- You can stage one ONIE firmware package for upgrade at a time.
- If you install an Enterprise SONiC image after staging a firmware package, when you reboot, only the Enterprise SONiC image is installed. To perform a firmware upgrade, stage firmware package again.
- If ONIE does not reboot the system due to an installation failure, the system does not boot into Enterprise SONiC automatically.

To install a firmware package, following procedure:

1. Use the copy command to copy a firmware file from a source location to the local file system or a network server.

```
copy sourcefilepath destinationfilepath
```

- *sourcefilepath*—Enter the source file path as the URL of a remote server using an FTP, HTTPPs, or SCP download.
 - `ftp://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]`
 - `http[s]://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]`
 - `scp://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]`
- *destination-path*—Enter the destination file path as a local directory (`home:filepath`) or a file path on an FTP, HTTP or HTTPS, SCP, or SSH server.

For example:

```
copy scp://userid:passwd@hostip/tftpboot/onie-firmware-x86_64-dellemc_z9400_c3758-
r0.3.51.5.1-17.tar home://onie-firmware-x86_64-dellemc_z9400_c3758-r0.3.51.5.1-17.tar
```

2. Use the image firmware install command to stage a firmware package.

```
image firmware install file-url
```

Enter the *file-url* location of the image in one for the following formats:

- `http[s]://hostip:/filepath`—Install the image from a remote HTTP or HTTPS server.

- `file://filepath`—Install the image from the local or a USB file system.

For example:

```
sonic# image firmware install file://home/admin/onie-update-full-x86_64-dell EMC_z9400_c3758-r0.3.51.5.1-17.tar
%Info: Check 'show image firmware status' for firmware package staging progress.
```

- To view the firmware staging status, use the `show image firmware status` command. The command output should display the Global operation status : `GLOBAL_STATE_SUCCESS` message. The following global operation statuses may be displayed:

- `GLOBAL_STATE_DOWNLOAD` during the firmware package downloading process
- `GLOBAL_STATE_STAGE_PROGRESS` during the firmware package staging process
- `GLOBAL_STATE_SUCCESS` after a successful staging of the firmware package

```
sonic# show image firmware status
%Info: System reboot is required to initiate the firmware upgrade operation.
%Info: Reboot will take longer than normal and the upgrade process should not be interrupted.
%Info: Device may auto-reboot during the upgrade process based on the components being upgraded.
%Info: After successful upgrade, the device will boot into SONiC.
-----
Global operation status : GLOBAL_STATE_SUCCESS
-----
File operation status   : TRANSFER_STATE_SUCCESS
File size(bytes)       : 79237120
File transfer bytes    : 79237120
File download speed    : 77380 KB/s
File progress          : 100%
Transfer start time   : 2023-09-25 05:16:38+0000
Transfer end time      : 2023-09-25 05:16:39+0000
-----
Stage operation status : STAGE_STATE_SUCCESS
Stage start time       : 2023-09-25 05:16:39+0000
Stage end time         : 2023-09-25 05:16:39+0000
-----
```

- `GLOBAL_STATE_FAILED` if you incorrectly specified the image file or entered the wrong server IP address; for example:

```
sonic# image firmware install http://10.1.1/tftpboot/S5200-BIOS-3.40.0.9-11.bin
%Error: File is not a valid ONIE firmware package
```

- To install the firmware package, reboot the switch.

```
sonic# reboot
reboot in process .....
```

View the pending firmware upgrade and the result of an earlier firmware upgrade

After a successful staging of a firmware package and before reboot:

```
sonic# show image firmware
Pending Firmware Upgrade(s)
=====
Name                                     Version        Date
-----
onie-update-full-x86_64-dell EMC_z9400_c3758-r0.3.51.5.1-15.tar  3.51.5.1-15
2023-09-25 05:16:39

Past Firmware Upgrade(s)
=====
Name                                     Version        Result
-----
```

After you reboot the switch, and the firmware upgrade has been performed in ONIE:

```
sonic# show image firmware
Pending Firmware Upgrade(s)
=====
Name                                     Version      Date
-----
Past Firmware Upgrade(s)
=====
Name                                     Version      Result
Date
-----
onie-update-full-x86_64-dell EMC_z9400_c3758-r0.3.51.5.1-15.tar    3.51.5.1-15
Success   2023-09-25 06:02:15
```

Cancel a pending firmware upgrade

To cancel a pending firmware upgrade, use the following command:

```
sonic# image firmware cancel
```

Install a software patch

Starting in Release 4.1.0, to install bug fixes and security upgrades in an operational network, it is no longer necessary to upgrade to a new Enterprise SONiC image. You can install software patches to update the current image that is running on a switch. Each patch makes changes to targeted system files. There is no need to reboot the switch because a patch applies software changes by restarting specific system services.

Usage notes

- A patch does not introduce new software features or modify configuration parameters so that no database migration is required.
- If more than one container is affected, the patch specifies the order in which containers are restarted.
- A patch automatically performs a system reboot or container restart, if required.
- The maximum number of patches that you can apply to an Enterprise SONiC image is three. To display the list of patches already applied or installed on the switch, use the `show image patch list` command.
- During patch installation, if a patch fails to install properly, any part that was applied is removed and the image is restored to its previous state.
- After a patch is applied to an image, the name of the Enterprise SONiC version is updated with the suffix `-patched-version`. For example, version `4.2.0-Enterprise_Premium` is updated to `4.2.0-Enterprise_Premium_Build163`; for example:

```
sonic# show version
Software Version : 4.2.0-Enterprise_Premium_Build163
Product          : Enterprise SONiC Distribution by Dell Technologies
Distribution     : Debian 10.13
Kernel           : 5.10.0-21-amd64
Config DB Version: version 4.2.1
Build Commit     : 94a714d2af
Build Date       : Thu Sep 28 23:36:00 UTC 2023
Built By         : sonicbld@sonic-lvn-csg-001
Platform         : x86_64-dell EMC_s5248f_c3538-r0
HwSKU            : DellEMC-S5248f-P-25G-DPB
ASIC              : broadcom
Hardware Version: X01
Serial Number    : CN046MRJCES0085N0005
Uptime            : 21:10:16 up 4 days, 43 min, 1 user, load average: 2.60, 2.41, 2.38
Mfg               : DELL EMC

REPOSITORY        TAG                               IMAGE ID          SIZE
docker-database  4.2.0-Enterprise_Premium_Build163  2c04cae1cd78  405MB
```

```

docker-database           latest          2c04cae1cd78    405MB
docker-dhcp-relay-ent-advanced 4.2.0-Enterprise_Premium_Build163  afb3662f3214 482MB
docker-dhcp-relay-ent-advanced latest          2c04cae1cd78    482MB
docker-eventd              latest          489449ab80b0    399MB
docker-eventd              latest          489449ab80b0    399MB
docker-fpm-frr              latest          e46b2af3eb1b    505MB
docker-fpm-frr              latest          e46b2af3eb1b    505MB
...
docker-snmp                latest          fff0aa178858    421MB
docker-snmp                latest          fff0aa178858    421MB
docker-sonic-mgmt-framework 4.2.0-Enterprise_Premium_Build163  cc855a082af8 641MB
docker-sonic-mgmt-framework latest          cc855a082af8 641MB
docker-sonic-telemetry      latest          1aeb2ec29950   583MB
docker-sonic-telemetry      latest          1aeb2ec29950   583MB
docker-stp                  latest          f18f134e9eb5    482MB
docker-stp                  latest          f18f134e9eb5    482MB
docker-swss-brcm-ent-advanced 4.2.0-Enterprise_Premium_Build163  9a914f59c7af 474MB
docker-swss-brcm-ent-advanced latest          9a914f59c7af 474MB
docker-syncd-brcm-ent-advanced 4.2.0-Enterprise_Premium_Build163  2b09920ca348 836MB
docker-syncd-brcm-ent-advanced latest          2b09920ca348 836MB
docker-tam                  latest          b2f192404790   485MB
docker-tam                  latest          b2f192404790   485MB
docker-teamd                latest          7fac033916f4    478MB
docker-teamd                latest          7fac033916f4    478MB
docker-udld                 latest          fd17f29375ac   485MB
docker-udld                 latest          fd17f29375ac   485MB
docker-vrrp                 latest          49dd5c0c47b9    486MB
docker-vrrp                 latest          49dd5c0c47b9    486MB

```

```

Applied Patch list:
07.12.22-0023-patch-framework-verification-patch 2019.02.16-07:55:25
docker-teamd

```

View patch updates

To view the list of patches that are already applied or installed on the switch, use the `show image patch list` command.

```

sonic# show image patch list
-----
Id  Tag                               Date             DepndsOn
-----
01  22.11.22-0001-patch-framework-verification-patch 2019.02.25-06:46:57

```

To view the patches that have been applied and removed (rolled back) for the current running image, use the `show image patch history` command.

```

sonic# show image patch history
-----
Id  Tag                               State  Status  Start            End
-----
01  22.11.22-0001-patch-framework-verification-patch  apply   complete 2019.02.25-06:44:44 2019.02.25-06:47:01
01  22.11.22-0001-patch-framework-verification-patch  rollback  complete 2019.02.25-06:36:17 2019.02.25-06:37:50
01  22.11.22-0001-patch-framework-verification-patch  apply   complete 2019.02.25-06:29:44 2019.02.25-06:31:56

```

Install a software patch

To install a patch, use the `image patch install {file:local-path | file-url}` command; for example:

```

sonic# image patch install file://home/admin/sonic-broadcom-enterprise-
advanced.bin.01.patch
%Info: Check 'show image patch status' for patch install progress.

```

To remove an installed patch, enter the `image patch rollback patch-tag-name` command, where `patch-tag-name` is the Tag value displayed in `show image patch list` output; for example:

```

sonic# image patch rollback "22.11.22-0001-patch-framework-verification-patch"
%Info: Check 'show image patch status' for patch rollback progress.

```

Any containers that have been patched are rolled back to their previous status. The image version is also restored back to its installed or previous version name.

To view the progress of a patch installation or removal, use the `show image patch status` command.

```
sonic# show image patch status
-----
Global operation status : GLOBAL_STATE_SUCCESS
-----
File operation status   : TRANSFER_STATE_SUCCESS
File size(bytes)       : 2665197550
File transfer bytes    : 2665197550
File progress          : 100%
Transfer start time   : 2022-11-22 06:19:42+0000
Transfer end time      : 2022-11-22 06:20:07+0000
-----
Install operation status : INSTALL_STATE_SUCCESS
Install start time     : 2022-11-22 06:21:06+0000
Install end time        : 2022-11-22 06:24:00+0000
```

Using USB storage media

Most Enterprise SONiC switches support a USB slot that can be used by an administrator to use removable mass storage media. The attached storage media can be used to store files, such as switch configuration, firmware images, and provisioning scripts. The files stored on USB media can be used for Zero Touch Provisioning (ZTP), troubleshooting switch operation, or during normal switch operation. An Enterprise SONiC NOS image stored on the USB media can also be used by the ONIE loader to install an image on a new switch from the factory.

Use MF-CLI commands to:

- Identify attached USB storage media and access their contents.
- Automatically mount USB devices.
- Manually mount and unmount the USB devices in order to access their contents.

Only users with an admin role users can write to the mounted USB device. All other users have read-only access.

An Enterprise SONiC switch supports vfat, ext2, ext3, and ext4 file systems on USB media.

Automatically detect and mount USB media

The file path where a USB device is mounted is automatically assigned in the format, `/media/usb#`, where `#` is between 0 and 7 for the number of an installed USB device.

- To enable the auto-detection of USB media and automatically mount all USB partitions which have a supported file system:

```
sonic(config)# usb enable
```

To disable USB auto-detection and unmount all USB partitions, enter the `no usb enable` command.

- To manually mount all available USB partitions with a supported file system to the corresponding mount points in the `/media` directory:
 1. Enable the auto-detection of USB media.

```
sonic(config)# usb enable
```

2. Mount the available USB partitions to the corresponding mount points in the `/media` directory:

```
sonic# usb mount
```

To unmount all USB mount points, enter the `usb un-mount` command.

i | NOTE: You must first enter the `usb enable` command before you can mount USB partitions using the `usb mount` command.

View USB contents

Use the `show usb` command to display information about USB device partitions which have a supported file system and which have been mounted.

```
sonic# show usb
Auto Detection: Enabled
-----
Mount Dir      Device Name  File System
```

```
-----  
/media/usb0  /dev/sdb1    ext2  
/media/usb1  /dev/sdb2    vfat
```

- Auto Detection — Displays USB auto-detection mode: enabled or disabled.
- Mount Dir — Directory where the USB device partition is mounted.
- Device Name — Block device that corresponds to the USB device partition.
- File System — Type of file system detected on the partition.

View USB device partitions

Use the `show usb partitions` command to display information about the detected USB device partitions.

```
sonic# show usb partitions  
-----  
Device Name  Mount Dir      File System  
-----  
/dev/sdb1    /media/usb0    ext2  
/dev/sdb2    /media/usb1    vfat  
/dev/sdb3
```

- Device Name — Block device that corresponds to the USB device partition.
- Mount Dir — Directory where the USB device partition is mounted. If the partition has not been mounted, the field is blank.
- File System — Type of file system detected on the partition. If the file system is not supported, the field is blank.

View USB devices

Use the `show usb devices` command to display information about inserted USB devices.

```
sonic# show usb devices  
-----  
USB Device   Manufacturer   Model Name  
-----  
/dev/sdb     Memorex       USB Flash Drive
```

- USB Device — Block device that corresponds to the detected USB device.
- Manufacturer — Manufacturer of the detected USB device.
- Model Name — Model of the detected USB device.

View USB mount point

Use the `dir usb#: /` command to display information about a mounted USB devcie and its file contents, where # is between 0 and 7 for the number of an installed USB device.

```
sonic# dir usb#: /  
  
sonic# dir usb0:/  
-----  
Date(Last Modified)  Size(Bytes)  Type   Filename  
-----  
2022-03-01 22:49      14798      -      config_db.json
```

Delete files in a USB mount point

Use the `delete usb#: //filename` command to delete a specified file in a mounted USB device.

```
sonic# delete usb#: //filename
```

For example:

```
sonic# delete usb0:/config_db.json
```

Copy files in a USB mount point

Use the `copy usb#: //filename` command to:

- Copy a specified file in a mounted USB device to a destination directory.

- Copy a file in a specified path to a mounted USB device.

```
sonic# copy usb#://filename destinationfilepath
```

Or

```
sonic# copy sourcefilepath usb#://filename
```

For example:

```
sonic# copy usb0://config_db.json home://config_db.json
```

```
sonic# copy http://10.1.1.1/v10/config_db.json usb0://config_db.json
```

System file management

Using SONiC file system commands, you can manage files in the SONiC operating system. An `admin` user role is required to perform these tasks:

- Copy a backup of the running and startup configuration to a remote server.
- Restore a system configuration file from a remote server.
- Copy core dump files to a remote server.
- Copy support bundle files to a remote server.
- Copy the event profile from a remote server or local storage.
- Copy log files to a remote server or local storage.

Copy a system file

Use the `copy` command to copy a file from a source to a destination location. Either the source or the destination should be a local file.

```
sonic# copy source-url destination-url
```

The exact format of the source and destination URLs varies according to the file or directory location. You can enter any of the following keyword aliases for a local system file or local file directory:

- `startup-configuration` — Configuration file applied at system startup.
- `running-configuration` — Current running configuration file.
- `config:` — System config directory.
- `home:` — \$HOME directory of the current user.
- `coredump:` — Coredump directory.
- `tech-support:` — Support Bundle directory.
- `log:` — Log directory.
- `event-profile:` — Event profile directory.

To specify a remote file or remote file directory:

- `ftp://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]` — Source or destination URL of an FTP server.
- `http://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]` — Source or destination URL of an HTTP or HTTPS server.
- `scp://[userid[:passwd]@]{hostname | host-ip}/directory-path/[filename]` — Source or destination URL of a Secure Shell (SSH) server.

Examples

To download a startup-config file from an FTP server:

```
sonic# copy ftp://admin:admin@10.10.10.10/new-startup-config.cfg startup-config
```

In an FTP copy operation, you can hide the password, and even the user ID, in the command syntax:

```
sonic# copy ftp://admin@10.10.10.10/new-startup-config.cfg startup-config  
Password: *****
```

Or

```
sonic# copy ftp://10.10.10.10/new-startup-config.cfg startup-config  
Username: admin  
Password: *****
```

To back up the startup-config file by creating a local file copy:

```
sonic# copy startup-config home://config-backup-AUG0821
```

To back up the running-config file by creating a local file copy:

```
sonic# copy running-config home://config-backup-AUG0821
```

To back up the startup-config file on an FTP server:

```
sonic# copy startup-config ftp://admin:admin@10.10.10.10/startup-config-bkup
```

To back up the running-config file on an FTP server:

```
sonic# copy running-config ftp://admin:admin@10.10.10.10/config-bkup
```

To back up the running-config file on an SSH server:

```
sonic# copy startup-config scp://admin:admin@10.10.10.10/config-bkup
```

In an SCP copy operation, you can hide the password, and even the user ID, in the command syntax:

```
sonic# copy startup-config scp://admin@10.10.10.10/config-bkup  
Password: *****
```

Or

```
sonic# copy startup-config scp://10.10.10.10/config-bkup  
Username: admin  
Password: *****
```

To restore a running-config file from an HTTP server:

```
sonic# copy http://inter128.acme.com/config-bkup running-config
```

To copy a SONiC coredump file to an FTP server:

```
sonic# copy coredump://sonic_dump_sonic_20210717_202813.tar.gz ftp://  
admin:admin@10.10.10.10/sonic_dump_sonic_20210717_202813.tar.gz
```

View system files

Use the `dir` command to view system files in a specified directory.

```
sonic# dir [home: | coredump: | tech-support: | logs: | config: | event-profile:]
```

Example

To view the contents of the `config:` directory:

```
sonic# dir config://  
-----
```

Date (Last Modified)	Size (Bytes)	Type	Filename
2023-09-29 00:53	41	-	asic_config_checksum
2023-09-29 15:58	0	s	ccd_mgmt_sock
2023-06-19 10:45	4096	d	cert
2023-09-29 15:58	334276	-	config_db.json
2020-09-17 01:48	671	-	config_db_version_registry.json
2023-09-29 00:56	1456	-	constants.yml
2023-09-29 15:58	12984	-	copp_config.json
2023-09-29 00:56	91	-	copp_feat_trap.json
2023-09-29 15:58	6	-	copp_rx_rate
2023-09-29 00:56	403	-	core_analyzer.rc.json
2023-09-29 00:56	2466	-	docker_limits.json
2023-09-29 15:59	4096	d	frr
2023-09-29 01:00	755	-	generated_services.conf
2023-09-29 00:55	65	d	hamd
2023-09-29 01:00	42	d	hw_resources
2023-09-29 15:58	5273	-	init_cfg.json
2023-10-02 11:36	334276	-	leaf2.json
2022-07-11 20:00	4096	d	licenses
2023-09-29 15:58	4096	d	old_config
2023-09-29 15:55	488884	-	
prev_config_db_version_registry.json-2023-09-29-035850			
2023-08-22 13:55	209	-	snmp.yml
2023-09-29 15:58	419	-	sonic_branding.yml
2023-09-29 15:58	134	-	sonic-environment
2023-09-29 15:58	330	-	sonic_version.yml
2023-09-29 00:56	14	-	updategraph.conf

Delete system files

Use the `delete` command to delete a system file from a specified directory.

```
sonic# delete {home://filename | coredump://filename | tech-support://filename | logs://filename | config://filename | event-profile://filename}
```

Example

To delete the `config_backup.json` file from the `home:` directory:

```
sonic# delete home://config_backup.json
```

Configure user login lockdown

Use the login lockdown feature to protect a switch from an intruder's continuous attempts to log in with an invalid password. The login lockdown feature allows you to reconfigure the default settings that are used to lock out user accounts after consecutive failed login attempts.

By default, the login lockdown feature is disabled. A maximum of three consecutive failed password attempts are supported on a switch. No lockdown period (0 minutes) is configured. To enable the login lockdown feature, configure a lockdown period.

Login failures include failures on the console as well. To allow locked out users to log in to the switch from the console, use the `login lockdown console-exempt` command.

```
sonic(config)# login lockdown {max-retries number | period minutes | console-exempt}
```

- `max-retries number` — Maximum number of consecutive failed login attempts that are allowed before a user is locked out (0 to 16; default 3).
- `period minutes` — Number of minutes that a user ID is prevented from logging in to the switch after the maximum number of failed login attempts is exceeded (0 to 43200; default 0).
- `console-exempt` — (Optional) Disable lockdown for console logins and enable a user to log in through the console when the user ID is locked out.

Configuration notes

- Dell Technologies recommends that you configure the lockout period as a nonzero value. If you set the `period minutes` value to zero, no lockout period is configured. As a result, any number of failed login attempts do not lock out a user.
- When a user is locked out after exceeding the maximum number of failed login attempts, other users can still log in to the switch.
- You can disable lockout for logins from the console by enabling the `login lockout console-exempt` option. This option also allows locked out users to log in to the switch through the console.
- To disable the login lockout feature, configure the lockout period as 0.
- To view the current login and lockout configuration, use the `show running-configuration` command.

```
sonic(config)# login lockout max-retries 5
sonic(config)# login lockout period 10
sonic(config)# login lockout console-exempt

sonic# show running-configuration | grep login
login lockout period 10
login lockout max-retries 5
login lockout console-exempt
```

Interface naming modes

Enterprise SONiC allows you to use three interface-naming modes: native, standard, and standard extended. By default, the switch is in native mode:

```
sonic# show interface-naming
Interface naming is set to native
```

When you configure standard mode or standard extended mode:

```
sonic# show interface-naming
Interface naming is set to standard
```

```
sonic# show interface-naming
Interface naming is set to standard extended
```

 **NOTE:** Dell Technologies recommends that you use the standard or standard extended interface-naming mode instead of the default native mode.

Native interface-naming mode

By default, an Ethernet interface is identified in *native* mode in the command-line interface as `Ethernet number` or `Ethernet number` — with or without a blank space; for example:

```
sonic(config)# interface Ethernet 4
sonic(conf-if-Ethernet4) #
```

Or

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4) #
```

In native interface-naming mode, Ethernet interfaces display in show outputs as:

```
sonic# show vlan
Q: A - Access (Untagged), T - Tagged
NUM      Status      Q Ports      Autostate  Dynamic
2        Inactive   A Ethernet8    Enable     No
                  T Ethernet18
                  T Ethernet36
```
sonic# show interface status

```

| Name      | Description | Oper | Reason       | AutoNeg | Speed | MTU  | Alternate Name |
|-----------|-------------|------|--------------|---------|-------|------|----------------|
| Ethernet0 | -           | down | admin-down   | off     | 40000 | 9100 | Eth1/1         |
| Ethernet4 | -           | down | err-disabled | off     | 40000 | 9100 | Eth1/2         |
| ...       |             |      |              |         |       |      |                |

In native interface-naming mode:

- In gNMI and REST API operations, you must enter Ethernet interfaces as `Ethernetnumber` with no blank space.
- You can enter Ethernet interfaces in abbreviated formats using lowercase or uppercase. For example, the CLI accepts these entries for Ethernet0:

```
E0
et0
Eth0
Eth 0
```

The system converts supported interface-name entries into `Ethernet0`.

## Standard interface-naming mode

To easily identify the front-panel port associated with an interface, enable *standard* interface-naming mode. In standard mode, non-breakout interfaces and breakout subinterfaces are identified in the CLI and displays in the format `Ethslot/port[/breakout-port]`; for example:

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2) #
```

```
sonic(config)# interface Eth1/2/4
sonic(conf-if-Eth1/2/4) #
```

In standard interface-naming mode, Ethernet interfaces display in show outputs as:

```
sonic# show vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
2 Inactive A Eth1/3 Enable No
 T Eth1/5/2
 T Eth1/10
...
sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/1 -
Eth1/2 -
...
sonic#
```

In standard interface-naming mode:

- Interface names that are entered in native mode are not supported in the CLI.
- You can enter Ethernet interfaces in abbreviated formats. For example, the CLI accepts the following entries for `Eth1/2`:

```
E1/2
e1/2
E 1/2
Eth 1/2
Et 1/2
```

The system converts supported short interface-name entries into `Eth1/2`.

- REST API requests using a curl command require interface names to be entered in standard interface format: `Eth$slot/$port[/breakout-port]`. You must replace each backward slash / in a standard interface name with %2F for ports in breakout and non-breakout modes. For example, enter `Eth1/2/4` as:

```
curl -X GET "https://ip-address/restconf/data/openconfig-interfaces:interfaces/
interface=Eth1%2F2%2F4" -H "accept: application/yang-data+json" -k -u "admin:admin"
```

### **Enable standard interface naming**

```
sonic(config)# interface-naming standard
```

To return to the default native interface naming:

```
sonic(config)# no interface-naming standard
```

After you change the interface-naming mode from native to standard or from standard to native, you are prompted to restart your Enterprise SONiC session:

```
Broadcast message: Interface naming mode has changed. Users running 'sonic-cli' are
required to restart your session.
sonic(config)# end
sonic#
sonic# exit
admin@sonic:~$
admin@sonic:~$ sonic-cl
sonic#
```

## Standard Extended interface-naming mode

Starting in 4.1.0 and later releases, you can use standard extended mode. Although *standard* port naming applies only to CLI configuration commands, REST API requests, gNMI remote procedure calls, Syslog messages, and SNMP views, *standard extended* port naming extends the standard mode to:

- Applications — Entering port syntaxes in FRR commands and other SONiC-specific applications
- Databases — Internal databases such as `config_db`
- Kernel name and kernel alias : Entering port syntaxes in Linux commands, such as `ifconfig` and `tcpdump`

The differences in port syntax naming between the three interface-naming modes are shown in these tables for a breakout and non-breakout port.

**Table 5. Non-breakout port naming**

| <b>Non-breakout port display in:</b> | <b>Native mode</b> | <b>Standard mode</b> | <b>Standard Extended mode</b> |
|--------------------------------------|--------------------|----------------------|-------------------------------|
| Databases                            | Ethernet0          | Ethernet0            | Eth1/1                        |
| Applications                         | Ethernet0          | Ethernet0            | Eth1/1                        |
| Port alias                           | Eth1/1             | Eth1/1               | Ethernet0                     |
| Kernal name                          | Ethernet0          | Ethernet0            | E1_1                          |
| Kernal alias                         | Ethernet0          | Ethernet0            | Eth1/1                        |
| CLI, REST API, gNMI                  | Ethernet0          | Eth1/1               | Eth1/1                        |
| Syslog                               | Ethernet0          | Eth1/1               | Eth1/1                        |
| SNMP                                 | Ethernet0          | Eth1/1               | Eth1/1                        |

**Table 6. Breakout port naming**

| <b>Breakout port display in:</b> | <b>Native mode</b> | <b>Standard mode</b> | <b>Standard Extended mode</b> |
|----------------------------------|--------------------|----------------------|-------------------------------|
| Databases                        | Ethernet48         | Ethernet48           | Eth1/49/1                     |
| Applications                     | Ethernet48         | Ethernet48           | Eth1/49/1                     |

**Table 6. Breakout port naming (continued)**

| Breakout port display in: | Native mode | Standard mode | Standard Extended mode |
|---------------------------|-------------|---------------|------------------------|
| Port alias                | Eth1/49/1   | Eth1/49/1     | Ethernet48             |
| Kernal name               | Ethernet48  | Ethernet48    | E1_49_1                |
| Kernel alias              | Ethernet48  | Ethernet48    | Eth1/49/1              |
| CLI, REST API, gNMI       | Ethernet48  | Eth1/49/1     | Eth1/49/1              |
| Syslog                    | Ethernet48  | Eth1/49/1     | Eth1/49/1              |
| SNMP                      | Ethernet48  | Eth1/49/1     | Eth1/49/1              |

**Enable standard extended interface naming**

```
sonic(config)# interface-naming standard extended
```

To return to the default native interface naming:

```
sonic(config)# no interface-naming standard
```

After you change the interface-naming mode from native to standard extended or from standard extended to native, you are prompted to restart your Enterprise SONiC session:

```
Broadcast message: Interface naming mode has changed. Config save followed by system
reload or reboot is required.
sonic(config)# exit
sonic# write memory
sonic# reboot
```

## Role-based access control

Role-based access control (RBAC) provides control for access and authorization. Users are granted permissions based on defined roles—not on their individual system user ID. Assign user roles based on job functions to allow users appropriate system access.

 **NOTE:** When you assign a user role, you also create the user with the configured access permissions.

RBAC places limitations on each role's permissions to allow you to partition tasks. At login, a user role authenticates and authorizes the user and places the user in the appropriate mode (EXEC mode or Linux shell).

Enterprise SONiC supports four predefined roles — `admin`, `operator`, `secadmin`, and `netadmin`. You cannot create additional roles. Each user role assigns permissions that determine the commands that a user can enter, and the actions a user can perform. RBAC provides an efficient way to administer user rights. If a user role matches one of the allowed user roles for a command, command authorization is granted.

Many users can have the same role. You can assign a single or multiple roles to each user. The following roles and combination of roles are supported:

- `admin`
- `operator`
- `netadmin`
- `secadmin`
- `netadmin` and `secadmin`
- `netadmin` and `operator`
- `secadmin` and `operator`, and so on

**Table 7. Roles and access permissions**

| Roles    | Access permission                                                                               | Default shell                                                                                                              |
|----------|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| admin    | A user has full read/write access to system.                                                    | When you log in with an admin role, you are placed in the Linux shell. The prompt is <code>admin@sonic:~\$</code> .        |
| operator | A user has only read access to system.                                                          | When you log in with an operator role, you are placed in the Management Framework CLI. The prompt is <code>sonic#</code> . |
| secadmin | A user has access to all security-related commands in the system.                               | When you log in with a secadmin role, you are placed in the Management Framework CLI. The prompt is <code>sonic#</code> .  |
| netadmin | A user has access to the configuration features that manage traffic flowing through the switch. | When you log in with a netadmin role, you are placed in the Management Framework CLI. The prompt is <code>sonic#</code> .  |

## Create users and assign roles

To create a user and configure switch access, use the `username password role` command. Until you configure a user role with this command, the user cannot access the switch.

- Enter a username, password, and role.

```
sonic(config)# username username password password role role
```

- `username username` — Enter a text string (up to 32 alphanumeric characters; one character minimum)
- `password password` — Enter a text string (up to 32 alphanumeric characters; one character minimum.)
- `role role` — Enter a user role:
  - `admin` — Full access to all commands in the system, exclusive access to commands that change the file system, and access to the system shell. An administrator can create user IDs and assign roles.
  - `operator` — Access to EXEC mode to view the current configuration. An operator cannot modify configuration settings on a switch.
  - `secadmin` — Access to all security-related commands in the system. A secadmin user can view, change, and run commands specific to security features. Also the user is allowed to access system-level information. Users are not allowed to log in into the Linux shell.
  - `netadmin` — Access to the network-related configuration commands that manage traffic flowing through the switches.

**(i) NOTE:** To create a user and assign a role, you must log in with the `admin` role.

- To remove user access to the switch, enter the `no username username` command.
- To change a user role without removing switch access, reenter the `username username password password role role` command with the same password and the new role.

**(i) NOTE:** During an Enterprise SONiC upgrade or downgrade, locally configured users and their passwords and roles are properly migrated when installing a SONiC image using the `image install` command. The config migration scripts automatically migrate the `config_db.json`, `/etc/passwd`, `/etc/group`, `/etc/shadow`, `/etc/gshadow`, `/home/*`, and `/etc/sonic/cert/` directories, and `/var/spool/mail` files. However, if you reinstall Enterprise SONiC from ONIE, and manually migrate a configuration from one switch to another by copying and restoring a `config_db.json` file or by provisioning Enterprise SONiC using custom ZTP scripts, you must:

- Manually reconfigure the local users using the `username password role` command or programmatic interfaces. Remotely authenticated users whose credentials are authenticated by RADIUS, TACACS+, or LDAP are not affected.
- Manually re-create or restore the certificate and private key files used for the REST and/or gNMI telemetry servers.

**(i) NOTE:** When you run the `username` command on the switch, the configuration is directly written into the kernel. As a result, the `write memory` command is not required to keep it after a reboot. Running the `username` command causes

local user accounts to be retained after a write erase operation is performed. However, if you reset the factory default configuration using the `factory` command, the configured local credentials are lost and cannot be restored from backup.

```
sonic(config)# username tango password charlie role operator
```

## Create user with multiple roles

```
sonic(config)# username tango password charlie role secadmin,netadmin
```

**Log in as operator and enter configuration commands — not allowed as operator**

## **Log in as administrator and change user role**

## Delete user

```
sonic(config)# no username tango
```

## View local and remote users

To monitor and display information on the locally configured and remote users logged in to the switch, use the `show users` and `show users configured` commands.

### View local and remote users

```
sonic# show users
admin ttyS0 2020-12-03 00:11
admin pts/0 2020-12-03 00:17 (10.14.1.95)
```

Starting in Release 4.1.0, to monitor the locally configured users who are logged in to the switch, you must use the `show users configured` command. Locally configured users with the `username password role role` command are no longer displayed in `show running-configuration | grep user` output and in the `config_db.json` file.

```
sonic# show users configured

User Role(s)

JustAnything operator
Kenna operator
admin admin
babuji admin
dellradius15 secadmin,operator
isg-admin operator
ldapuser admin
localprocessdmzscan operator
processdmzscan operator
t088788 admin
t098888 admin
tacacsadmin operator
tacacsuser operator
```

## Authentication, authorization, and accounting

Authentication, authorization, and accounting (AAA) services secure networks against unauthorized access. Besides local authentication, Enterprise SONiC supports these client/server authentication systems:

- RADIUS (remote authentication dial-in user service)
- TACACS+ (terminal access controller access control system)
- LDAP (lightweight directory access protocol).

For RADIUS and TACACS+ authentication, a switch acts as a client and sends authentication requests to a server that contains all user authentication and network service access information.

**(i) NOTE:** In RADIUS server authentication, Enterprise SONiC does not interoperate with Microsoft Windows Network Policy Server (NPS). NPS does not support RFC 5607 to provide the Management-Privilege-Level (MPL) attribute. Enterprise SONiC requires the MPL attribute with value 15 to grant read-write access to an authorized user.

AAA configuration consists of setting up user access controls:

1. Configure the authentication methods used to allow access to the switch — see [Configure authentication](#).
2. Configure the level of command authorization for authenticated users — see [Role-based access control](#).

To view the security information collected for user sessions, see [Audit log](#).

## Configure authorization

AAA command authorization controls user access to a set of commands assigned to users and is performed after user authentication. When enabled, AAA authorization checks a remote authorization server for each command that a user enters on the switch. If the commands that are entered by the user are configured in the remote server for the user, the remote server authorizes the usage of the command.

### Configuration notes

**(i) NOTE:** Enterprise SONiC supports only TACACS+-based server authorization.

- Command authorization is performed only after a user is authenticated — see [Configure authentication](#).
- By default, the role you configure with the `username password role` command sets the level of CLI commands that a user can access. If you also enable authorization, command authorization for an authenticated user is first checked before role-based access is applied.
- AAA authorization uses the TACACS+ servers configured for authentication — see [Configure TACACS+ servers](#). The server priority and timeout configured with the `tacacs-server host` command are used. For detailed information about how to configure vendor-specific attributes on a security server, see the respective RADIUS or TACACS+ server documentation.

### Enable TACACS+-based command authorization

- sonic(config)# aaa authorization commands default group tacacs+ local
  - `default group tacacs+ local` — Use the default method list with the TACACS+ group of servers configured with the `tacacs-server host` command for command authorization. If none of the configured TACACS+ servers are reachable, use local role-based (RBAC) authorization to authorize commands.

### Verify TACACS+-based authorization

```
sonic# show aaa

AAA Authentication Information

failthrough : True
login-method : tacacs+, local

AAA Authorization Information

commands : tacacs+, local
```

## Configure authentication

AAA authentication verifies and grants user access to the switch. You can configure authentication to use the local username/password database or one or more remote RADIUS and TACACS+ servers or LDAP.

### Authentication methods

A switch uses a list of authentication methods to define the types of authentication and the sequence in which they apply. By default, only the `local` authentication method is used to authenticate users with the local user database. You can also configure TACACS+, RADIUS, or LDAP as the primary or secondary authentication method with local authentication. You can specify only one remote authentication service — TACACS+ or RADIUS or LDAP.

- To use the group of configured TACACS+ servers, enter `group tacacs+` — see [Configure TACACS+ servers](#).
- To use the group of configured RADIUS servers, enter `group radius` — see [Configure RADIUS servers](#).
- To use the group of configured LDAP servers, enter `group ldap` — see [Configure LDAP](#).

```
sonic(config)# aaa authentication login default
{{[group {{[ldap [local]]} | {[radius [local]]} | {[tacacs+ [local]]}}]} | {[local {[group {[ldap] | [radius] | [tacacs+]}]}]}}
```

The authentication methods in the method list are run in the order you enter them. Re-enter the methods to replace and change the order in which the authentication methods are applied; for example:

```
sonic(config)# aaa authentication login default radius local
sonic(config)# aaa authentication login default local group ldap
```

To remove the configured authentication methods and return to only local authentication, enter the `no aaa authentication login default` command.

You must configure a TACACS+, RADIUS, or LDAP server correctly and ensure that the connectivity to the server is available through the Management interface. If you configure remote authentication using a TACACS+ or RADIUS server, all user logins are authenticated by the server. If the authentication fails, AAA checks the `failthrough` configuration and authenticates the user based on the local database if fail-through is enabled.

## View AAA authentication

```
sonic# show aaa

AAA Authentication Information

failthrough : False
login-method : local, tacacs+
...
```

## Enable fail-through for TACACS+ and RADIUS authentication

Use the fail-through option if you configure TACACS+- or RADIUS-based authentication with more than one remote server. The fail-through feature continues to access each server in the method list if an authentication request fails on one server. Enable or disable authentication fail-through with the [no] aaa authentication failthrough command. Authentication fail-through is enabled by default.

```
sonic(config)# aaa authentication failthrough {enable | disable}
```

## Fail-through authentication scenarios

By default, fail-through for TACACS+, RADIUS, and LDAP authentication is disabled. In this case, TACACS+, RADIUS, and LDAP-based authentication handle user logins in the following ways for Authentication failure and Authentication server not reachable returned messages.

 **NOTE:** In the Fail-through tables, admin refers to logging in with the username admin. "Local users" log in with a configured username and password.

**Table 8. Fail-through disabled: Authentication scenarios**

| Configured authentication methods    | Authentication failure returned from authentication server | Authentication server not reachable                    |
|--------------------------------------|------------------------------------------------------------|--------------------------------------------------------|
| TACACS+ and local user configuration | admin and local users are <b>not</b> allowed to log in     | admin and local users are allowed to log in            |
| Only TACACS+ authentication          | admin and local users are <b>not</b> allowed to log in     | admin and local users are <b>not</b> allowed to log in |
| RADIUS and local user configuration  | admin and local users are <b>not</b> allowed to log in     | admin and local users are allowed to log in            |
| Only RADIUS authentication           | admin and local users are <b>not</b> allowed to log in     | admin and local users are <b>not</b> allowed to log in |
| LDAP and local user configuration    | admin and local users are allowed to log in                | admin and local users are allowed to log in            |
| Only LDAP authentication             | admin and local users are <b>not</b> allowed to log in     | admin and local users are <b>not</b> allowed to log in |

If you enable fail-through for TACACS+, RADIUS, and LDAP authentication, user logins are handled in the following ways for Authentication failure and Authentication server not reachable returned messages.

**Table 9. Fail-through enabled: Authentication scenarios**

| Configured authentication methods    | Authentication failure returned from authentication server | Authentication server not reachable                    |
|--------------------------------------|------------------------------------------------------------|--------------------------------------------------------|
| TACACS+ and local user configuration | admin and local users are allowed to log in                | admin and local users are allowed to log in            |
| Only TACACS+ authentication          | admin and local users are <b>not</b> allowed to log in     | admin and local users are <b>not</b> allowed to log in |
| RADIUS and local user configuration  | admin and local users are allowed to log in                | admin and local users are allowed to log in            |
| Only RADIUS authentication           | admin and local users are <b>not</b> allowed to log in     | admin and local users are <b>not</b> allowed to log in |

**Table 9. Fail-through enabled: Authentication scenarios (continued)**

| Configured authentication methods | Authentication failure returned from authentication server | Authentication server not reachable                    |
|-----------------------------------|------------------------------------------------------------|--------------------------------------------------------|
| LDAP and local user configuration | admin and local users are allowed to log in                | admin and local users are allowed to log in            |
| Only LDAP authentication          | admin and local users are <b>not</b> allowed to log in     | admin and local users are <b>not</b> allowed to log in |

## Configure TACACS+ servers

You can configure remote TACACS+ servers for user authentication. If you use multiple TACACS+ servers or if a TACACS+ server returns an authentication failure, enable fail-through so that the AAA authentication passes to the next TACACS+ server in the list or to the secondary authentication method.

### Add TACACS+ server to authentication list

You can configure up to eight TACACS+ servers for remote user authentication using the `tacacs-server host` command. The corresponding TACACS+ entries in the SONiC PAM configuration file are updated.

When a user logs in, TACACS+ servers perform authentication in the order the servers are listed in the method list. When a TACACS+ server times out, the switch contacts the next server in the method list according to the configured priority value. For information about using the fail-through option for TACACS+-based authentication, see [Configure authentication](#).

```
sonic(config)# tacacs-server host {hostname | ip-address | ipv6-address} [port port-number[.subinterface]] [timeout seconds] [key text] [type authentication-type] [priority value] [vrf {mgmt | vrf-name}]
```

To configure a TACACS+ server, enter its hostname (63 characters maximum), IP, or IPv6 address and these optional values:

- TCP port number on the server (1 to 65535; default 49)
- Transmission timeout in seconds (1 to 60; default 5)
- Secret key text that is shared between a TACACS+ server and the switch (up to 32 characters)
- Authentication type — `chap`, `pap`, or `mschap`; default `pap`. The authentication algorithm is used to encrypt and decrypt data that is sent and received between the switch and the TACACS+ server.
- Priority used to access multiple TACACS+ servers to authenticate users (1 to highest priority 64; default 1)
- Enter a VRF name to specify the VRF to use to reach the TACACS+ server. Before you configure a non-default VRF to contact the TACACS+ server, you must first create the VRF — see [Configure nondefault VRF instances](#).

```
sonic(config)# tacacs-server host 1.1.1.1 port 11 timeout 10 key mykey type pap priority 11
```

### View configured TACACS+ servers

```
sonic# show tacacs-server host

HOST AUTH-TYPE KEY-CONFIG PORT PRIORITY TIMEOUT VRF

1.1.1.1 pap Yes 11 11 10 mgmt
2.2.2.2 mschap Yes 20 20 20 mgmt

sonic# show tacacs-server host 1.1.1.1

HOST AUTH-TYPE KEY-CONFIG PORT PRIORITY TIMEOUT VRF

1.1.1.1 pap Yes 11 11 10 mgmt
```

### Configure global TACACS+ authentication type

Configure a default TACACS+ authentication type that is used for remote TACACS+-based user authentication. If you do not specify a `type` value in the `tacacs-server host` command, the default value is used. Different authentication types use different access-request and access-challenge messages.

- `chap` — Challenge handshake authentication protocol
- `pap` — Password authentication protocol (default)

- mschap — Microsoft challenge handshake authentication protocol
- login — Microsoft challenge handshake authentication protocol

```
sonic(config)# tacacs-server auth-type {pap | chap | mschap | login}
```

### Configure global TACACS+ shared key

Configure a global shared secret key that is used by the switch as a TACACS+ client to authenticate itself on a TACACS+ server (up to 32 characters). Valid characters are 0 to 9, A to Z, and a to z. The global shared key is used only on TACACS+ authentication servers which were configured without a key *secret* value in the tacacs-server host command.

```
sonic(config)# tacacs-server key testing123
```

### Configure global TACACS+ source interface

Configure the source interface that is used by the switch to communicate with TACACS+ servers. By default, no source interface is configured.

```
sonic(config)# tacacs-server source-interface {Eth slot/port[/breakout-port] [.subinterface] | Loopback number | Management 0 | PortChannel number[.subinterface] | Vlan vlan-id}
```

### Configure global TACACS+ timeout

Configure a global timeout value for all TACACS+ servers that are used for remote authentication (1 to 60 seconds; default 5). The global timeout is used only on TACACS+ authentication servers which were configured without a specified timeout with the tacacs-server host command.

```
sonic(config)# tacacs-server timeout 60
```

### View global TACACS+ server settings

```
sonic# show tacacs-server global

TACACS Global Configuration

source-interface : Loopback0
timeout : 10
auth-type : chap
key configured : Yes
```

### Copy TACACS+ server configuration to another switch

When you copy a TACACS+ server configuration from one switch to another, take into account these requirements:

- The two switches must be configured with the same primary encryption key (PEK) that is used to encrypt protocol passwords. The primary encryption key may be the default or a user-configured value. Because the default primary encryption key is unique on each switch, you must configure a new primary encryption key on both switches before copying a TACACS+ configuration from one to the other.
- To view the primary encryption key that the switch uses, enter the show config-key password-encrypt command. In the command output, True indicates a user-configured PEK; False indicates the default PEK. For example:

```
sonic# show config-key password-encrypt
Primary encryption key configured: False
```

To configure a new primary encryption key, use the key config-key password-encrypt command:

```
sonic# key config-key password-encrypt
New key: <enter key>
Confirm key: <enter key>
```

- When you configure the primary encryption key, the configured protocol passwords — such as the global TACACS+ shared key — are automatically re-encrypted. Dell Technologies recommends that you configure the PEK during a maintenance time.
- After you copy a TACACS+ server configuration from one switch to another, if TACACS+ user authentication (or another protocol) does not work, the switches may be using different primary encryption keys. Verify the primary encryption keys on each switch.

**(i) NOTE:** A primary encryption key is hidden and cannot be displayed using the MF-CLI. Take extra care to note and remember any non-default primary encryption keys that you configure on the switch.

## Configure RADIUS servers

You can configure remote RADIUS servers for user authentication. If you use multiple RADIUS servers or if a RADIUS server returns an authentication failure, enable fail-through so that the AAA authentication process to the next RADIUS server in the list or to the secondary authentication method.

### Add RADIUS server to authentication list

You can configure up to eight RADIUS servers for remote user authentication using the `radius-server host` command. The corresponding RADIUS entries in the SONiC PAM configuration file are updated.

When a user logs in, RADIUS servers perform authentication in the order the servers are listed in the method list. When a RADIUS server times out, the switch contacts the next server in the method list according to the configured priority value. For information about how to use the fail-through option for RADIUS-based authentication, see [Configure authentication](#).

```
sonic(config)# radius-server host {hostname | ip-address | ipv6-address} [auth-port port-number] [auth-type authentication-type] [key text] [priority value] [retransmit number] [source-interface {Eth slot/port[/breakout-port][.subinterface] | Loopback number | Management 0 | PortChannel number[.subinterface] | Vlan vlan-id}] [timeout seconds] [vrf {mgmt | vrf-name}]
```

To configure a RADIUS server, enter its hostname (63 characters maximum), IP, or IPv6 address and these optional values:

- UDP port number on the server (1 to 65535; default 1812).
- Transmission timeout in seconds (1 to 60; default 5).
- Number of times a request for user authentication is resent to a RADIUS server (0 to 10; default 3).
- Secret key text that is shared between a RADIUS server and the switch (31 characters maximum). This key is encrypted by the system.
- Authentication type — chap, pap, or mschapv2; default pap; the authentication algorithm is used to encrypt/decrypt data that is sent and received between the switch and the RADIUS server.
- Priority used to access multiple RADIUS servers to authenticate users (1 to highest priority 64; default 1).
- Enter a VRF name to specify the VRF to use to reach the RADIUS server.

```
sonic(config)# radius-server host 1.1.1.1 port 11 timeout 10 key mykey type pap priority 11
```

### View configured RADIUS servers

```
sonic# show radius-server

RADIUS Global Configuration

timeout : 5
auth-type : pap
key configured : Yes

HOST AUTH-TYPE KEY-CONFIG AUTH-PORT PRIORITY TIMEOUT RTSMT VRF SI

1.1.1.1 - Yes 1812 - - - - -
Management0
```

### Configure global RADIUS authentication type

Configure a default RADIUS authentication type that is used for remote RADIUS-based user authentication. If you do not specify a `type` value in the `radius-server host` command, the default value is used. Different authentication types use different access-request and access-challenge messages.

- chap — Challenge handshake authentication protocol
- pap — Password authentication protocol (default)
- mschapv2 — Microsoft challenge handshake authentication protocol, Version 2

```
sonic(config)# radius-server auth-type {pap | chap | mschap}
```

## Configure global RADIUS shared key

Configure a global shared secret key that is used by the switch as a RADIUS client to authenticate itself on a RADIUS server (up to 65 characters). Valid characters are: ASCII printable except for SPACE, #, and comma. The global shared key is used only on RADIUS authentication servers for which the `key secret` value in the `radius-server host` command is not configured.

```
sonic(config)# radius-server key 23232
```

The system stores the keys that are used in the `radius-server key` and `radius-server host` commands in an encrypted format.

```
Sonic# show running-configuration | grep radius
radius-server timeout 5
radius-server key U2FsdGVkX1/ExhX5a44QLLMfA9mVYgs72bEI5aTET4g= encrypted
radius-server auth-type pap
radius-server host 1.1.1.1 auth-port 1812 key
U2FsdGVkX1+aefqOdzuREbmqMfolleWitDq4TssE8Q= encrypted
```

After you configure the key using the `radius-server key` and `radius-server host` commands and press the up arrow, the console does not display the password that you configured.

```
Sonic(config)# radius-server host 1.1.1.1 key 1111
Sonic(config)# radius-server host 1.1.1.1 key *****
Sonic(config)# radius-server key test1
Sonic(config)# radius-server key *****
```

## Configure per-host RADIUS source interface

Configure the source interface that is used by the switch to communicate with a RADIUS host server. By default, no source interface is configured.

```
sonic(config)# radius-server host {hostname | ip-address | ipv6-address} source-
interface {Eth slot/port[/breakout-port][.subinterface] | Loopback number | Management
0 | PortChannel number[.subinterface] | Vlan vlan-id}
```

## Configure global RADIUS timeout

Configure a global timeout value for all RADIUS servers that are used for remote authentication (1 to 60 seconds; default 5). The global timeout is used only on RADIUS authentication servers which were configured without a specified timeout with the `radius-server host` command.

```
sonic(config)# radius-server timeout seconds
```

## View global RADIUS server settings

```
sonic# show radius-server

RADIUS Global Configuration

timeout : 5
auth-type : pap
key configured : Yes

HOST AUTH-TYPE KEY-CONFIG AUTH-PORT PRIORITY TIMEOUT RTSMT VRF SI

1.1.1.1 - Yes 1812 - - - - -
```

## Authenticated user privilege levels

After successful RADIUS authentication, a user is assigned access privileges and enters the Linux shell or the MF-CLI level on the switch. The Management-Privilege-Level (MPL) attribute, which is returned from the RADIUS server, is used to grant different management access levels to authenticated users. An example of the RADIUS-authenticated user privileges that are returned to the switch:

**Table 10. RADIUS-authenticated user privileges**

| MPL (privilege) | Role assigned | Shell       |
|-----------------|---------------|-------------|
| 15              | admin         | Linux-Shell |

**Table 10. RADIUS-authenticated user privileges (continued)**

| MPL (privilege) | Role assigned | Shell  |
|-----------------|---------------|--------|
| 14              | netadmin      | MF-CLI |
| 13              | secadmin      | MF-CLI |
| 1 - 12          | operator      | MF-CLI |

#### Troubleshoot RADIUS servers

To debug RADIUS service, check the log files in the `/var/log/auth.log` folder, and the log files in `show in-memory-logging` and `show logging` SONiC CLI command output.

For more detailed logging, set the `debug` field to `True` in the `authentication` key of the AAA table in the `CONFIG_DB` redis database, or contact Technical Support.

## Dynamic Authorization Server

RFC 5176 discusses about Dynamic Authorization Server (DAS) and Dynamic Authorization Client (DAC).

The Dynamic Authorization Client sends two types of messages, namely the Disconnect Message and the Change of Authorization Message. The DAS acts on these messages and sends an acknowledgment (ACK) or a negative acknowledgment (NAK) for these messages.

The Disconnect Message from DAC may result in termination of a session of a user. The Change of Authorization message from a DAC results in a change of authorization status of the session.

As mentioned in the RFC 5176, there is no mechanism in the RADIUS protocol to send messages from a RADIUS server to a Network Access Server (NAS).

Enterprise SONiC implements Dynamic Authorization feature that includes the Dynamic Authorization Server (DAS) functionality as defined in RFC 5176 (Dynamic Authorization Extensions to Remote Authentication Dial In User Services).

 **NOTE:** DAS is available on N-series and E-series platforms.

### Disconnect messages

When Enterprise SONiC DAS receives a Disconnect Message from a DAC, it performs a basic validation and notifies the registered applications. The registered applications look for NAS identification, and User Identity attributes available in the Disconnect Message. If a match for the NAS attribute and the identity of the user are found, the system disconnects matching sessions and sends a response to DAS. Upon receiving the response from the registered applications, the DAS processes it and generates the final result. The final response (DM-ACK or DM-NAK) is sent to the DAC.

### Change of Authorization

The Change of Authorization (CoA) message is used to change session authorization. When Enterprise SONiC DAS receives a CoA message, the system performs a basic validation and notifies the registered applications. The registered applications read the request and match the NAS identification and Session Identity attributes before looking for additional attributes in the request. If NAS identification or Session identity attributes do not match, the system responds to DAS as failure with an error-code attribute as `Session-Context-NotFound`.

If the attributes match, the registered applications look for additional attributes and then process them accordingly. Once the attributes are processed, the response is sent to DAS. DAS upon receiving the response from the registered applications, process it and generates the final result. The final response (CoA-ACK or CoA-NAK) is sent to the DAC.

## Dynamic Authorization Server

1. Enable DAS functionality.

```
aaa server radius dynamic-author
```

- Configure the device to ignore a RADIUS server when it receives a `bounce-host-port` message.

```
authentication command bounce-port ignore
```

- Configure the device to ignore a RADIUS server when it receives a `disable-host-port` message.

```
authentication command disable-port ignore
```

- Specify the UDP port on which a device listens for requests from configured DACs.

```
Port port-number
```

- port-number* - Enter a port number. The range is from 1025 to 65535.

- Specify the type of authorization that the device must use for clients.

```
auth-type {any | all | session-key}
```

- `all` — Selects any COA client authentication type. Any authentication attribute may match for the authentication to succeed.
- `any` — Selects all COA client authentication types. All authentication attributes must match for the authentication to succeed.
- `session-key` — Indicates that the session-key must match for authentication to succeed.

- Configure the device to ignore the session key.

```
ignore session-key
```

This command fails to run when the authentication type using the `auth-type` command is set to `session-key` as the authentication can happen only based on the session-key attribute.

- Configure the device to ignore the server key.

```
ignore server-key
```

- Configure a global shared secret that is used for all dynamic authorization clients that do not have an individual shared secret key configured.

```
server-key key-string [encrypted]
```

- Configure the IP address, IPV6 address, or hostname of the client (Dynamic Authorization Client).

```
client {ip-address | ipv6-address | hostname} [server-key key-string]
```

- ip-address* — Specify the IP address of the client.
- ipv6-address* — Specify the IPv6 address of the client.
- hostname* — Specify the hostname of the client.
- `server-key` *key-string* — Encrypt the key string.

## View DAS information

View the dynamic authorization server parameters.

```
sonic# show radius-server dynamic-author

AdminMode..... Enabled
Port..... 1700
Auth Type..... any
Global Secret Key..... Yes
Ignore Server Key..... Disabled
Ignore Session Key..... Disabled
CoA Bounce Host Port..... Accept
CoA Disable Host Port..... Accept

Client Address Secret

```

|              |     |
|--------------|-----|
| 10.89.108.26 | No  |
| 1.1.1.1      | Yes |

View the DAS global and per client counters.

```
sonic# show radius-server dynamic-author statistics

Number of CoA Requests Received..... 5
Number of CoA ACK Responses Sent..... 2
Number of CoA NAK Responses Sent..... 3
Number of CoA Requests Ignored..... 1
Number of CoA Missing/Unsupported Attribute R.. 0
Number of CoA Session Context Not Found Reque.. 2
Number of CoA Invalid Attribute Value Request.. 0
Number of Administratively Prohibited Request.. 0

sonic# show radius-server dynamic-author statistics client all

DAC Address..... 10.89.108.26
Number of CoA Requests Received..... 4
Number of CoA ACK Responses Sent..... 0
Number of CoA NAK Responses Sent..... 4
Number of CoA Requests Ignored..... 0
Number of CoA Missing/Unsupported Attribute Requests.. 0
Number of CoA Session Context Not Found Requests..... 4
Number of CoA Invalid Attribute Value Requests..... 0
Number of Administratively Prohibited Requests..... 0

DAC Address..... 10.52.139.190
Number of CoA Requests Received..... 3
Number of CoA ACK Responses Sent..... 0
Number of CoA NAK Responses Sent..... 3
Number of CoA Requests Ignored..... 3
Number of CoA Missing/Unsupported Attribute Requests.. 0
Number of CoA Session Context Not Found Requests..... 0
Number of CoA Invalid Attribute Value Requests..... 0
Number of Administratively Prohibited Requests..... 0

sonic# show radius-server dynamic-author statistics client 10.89.108.26

DAC Address..... 10.89.108.26
Number of CoA Requests Received..... 4
Number of CoA ACK Responses Sent..... 0
Number of CoA NAK Responses Sent..... 4
Number of CoA Requests Ignored..... 0
Number of CoA Missing/Unsupported Attribute Requests.. 0
Number of CoA Session Context Not Found Requests..... 4
Number of CoA Invalid Attribute Value Requests..... 0
Number of Administratively Prohibited Requests..... 0
```

## Clear DAS information

Clear radius dynamic authorization global counters and per DAS client counters.

```
sonic# clear radius-server dynamic-author statistics [client { all | ipv4 | ipv6 | hostname }]
```

## Configure LDAP

Once you have configured the LDAP server host, you are now ready to configure LDAP.

```
ldap-server {{timelimit timelimit_val} | {bind-timelimit bind_timelimit_val} | {idle-timelimit idle_timelimit_val} | {retry retry_val} | {port port_val} | {scope scope_val} | {version ldap_version_val} | {base ldap_base_val} | {ssl ssl_val} | {binddn binddn_val} | {bindpw bindpw_val} | {pam-filter pam_filter_val} | {pam-login-attribute pam_login_val} | {pam-group-dn pam_group_val} | {pam-member-attribute pam_member_val} | {sudoers-base sudoers_val} | {nss-base-passwd nss_base_passwd_val} |
```

```
{nss-base-group nss_base_group_val} | {nss-base-shadow nss_base_shadow_val} | {nss-base-netgroup nss_base_netgroup_val} | {nss-base-sudoers nss_sudoers_val} | {nss-initgroups-ignoreusers nss-initgroups_val}}
```

- *timelimit\_val* — Time limit
- *bind\_timelimit\_val* — Bind time limit
- *idle\_timelimit\_val* — Idle time limit
- *retry\_val* — Retry
- *port\_val* — Port
- *scope\_val* — Scope; select sub or base
- *ldap\_version\_val* — LDAP version
- *ldap\_base\_val* — LDAP base
- *ssl\_val* — SSL; select on, off, or start\_tls
- *binddn\_val* — Distinguished name to bind
- *bindpw\_val* — Credentials to bind; valid characters include ASCII printable except space
- *pam\_filter\_val* — PAM filter name
- *pam\_login\_val* — PAM login attribute (default uid)
- *pam\_group\_val* — PAM group distinguished name
- *pam\_member\_val* — PAM member attribute value
- *sudoers\_val* — Sudo base distinguished name
- *nss\_base\_passwd\_val* — NSS search base password
- *nss\_base\_group\_val* — NSS search base group
- *nss\_base\_shadow\_val* — NSS search base for shadow map
- *nss\_base\_netgroup\_val* — NSS search base for netgroup map
- *nss\_sudoers\_val* — NSS search base for sudoers map
- *nss-initgroups\_val* — NSS initialization groups ignore users value

Specify the timeout interval for LDAP servers (1 to 60 seconds; default 5) in CONFIGURATION mode.

```
ldap-server timelimit timelimit_val
```

## Configuration

```
sonic(config)# ldap-server timelimit 13
sonic(config)# ldap-server bind-timelimit 10
sonic(config)# ldap-server idle-timelimit 12
sonic(config)# ldap-server retry 8
sonic(config)# ldap-server port 81
sonic(config)# ldap-server scope sub
sonic(config)# ldap-server version 2
sonic(config)# ldap-server base basetest
sonic(config)# ldap-server ssl on
sonic(config)# ldap-server binddn dnname
sonic(config)# ldap-server bindpw testpasswd
sonic(config)# ldap-server pam-filter testfilter
sonic(config)# ldap-server pam-login-attribute loginattrstring
sonic(config)# ldap-server pam-group-dn grpdn
sonic(config)# ldap-server pam-member-attribute attrstring
sonic(config)# ldap-server sudoers-base dnqrystr
sonic(config)# ldap-server nss-base-passwd dnsearchstr
sonic(config)# ldap-server nss-base-group grpmap
sonic(config)# ldap-server nss-base-shadow grpmap
sonic(config)# ldap-server nss-base-netgroup netgrpstr
sonic(config)# ldap-server nss-base-sudoers sudomap
sonic(config)# ldap-server nss-initgroups-ignoreusers grpstr
sonic(config)# no ldap-server nss-initgroups-ignoreusers grpstr
```

## Configure LDAP server hosts

To access a remote LDAP server, you must configure the IP address or the hostname of the LDAP server on the Enterprise SONiC device. You can configure up to eight LDAP servers.

```
ldap-server host host_val [use-type use_type_val] [port server_port_val] [priority priority_val] [ssl ssl_val] [retry retry_val]
```

To configure an LDAP server host, enter its hostname (up to 63 characters), IP, or IPv6 address and these optional values:

- *use\_type\_val* — (Optional) Use type; select all, nss, sudo, pam, nss\_sudo, nss\_pam, or sudo\_pam
- *server\_port\_val* — (Optional) Server port number
- *priority\_val* — (Optional) Port priority
- *ssl\_val* — (Optional) SSL; select on, off, or start\_tls
- *retry\_val* — (Optional) Retries

```
sonic(config)# ldap-server host 4.5.6.7 use-type nss port 300 priority 12 ssl on retry 5
```

### View configured LDAP servers

```
sonic# show ldap-server

LDAP Global Configuration

binddn : dc=sji,dc=example,dc=com

LDAP NSS Configuration

ssl : start_tls

HOST USE-TYPE PORT PRIORITY SSL RETRY
4.5.6.7 NSS - 1 START_TLS 2
```

## AAA with LDAP authentication

To configure an LDAP server for authentication, you first must setup AAA name services before you can configure the LDAP server settings.

### AAA configuration

1. Configure AAA to use LDAP for the login authentication method.

```
sonic(config)# aaa authentication login default group ldap
```

2. Configure AAA to use LDAP for the authorization login method.

```
sonic(config)# aaa authorization login default group ldap
```

3. Configure the AAA name-service settings.

**(i) NOTE:** LDAP name service is used by default if LDAP authentication is configured in AAA.

```
sonic(config)# aaa name-service passwd ldap
sonic(config)# aaa name-service shadow ldap
sonic(config)# aaa name-service group ldap
sonic(config)# aaa name-service netgroup ldap
sonic(config)# aaa name-service sudoers ldap
```

## LDAP server global configuration

1. Set the LDAP server time limit in seconds (0 to 65535; default 0).

```
sonic(config) # ldap-server timelimit timelimit
```

2. Set the LDAP server bind time limit in seconds (0 to 65535; default 10).

```
sonic(config) # ldap-server bind-timelimit bind-timelimit
```

3. Set the LDAP server idle time limit in seconds (0 to 65535; default 0).

```
sonic(config) # ldap-server idle-timelimit idle-timelimit
```

4. Set the LDAP server retry value in seconds (0 to 10; default 0).

```
sonic(config) # ldap-server retry retry
```

5. Set the LDAP server port number (0 to 65535; default 389).

```
sonic(config) # ldap-server port port
```

6. Set the LDAP server scope; select sub, one, or base; default sub.

```
sonic(config) # ldap-server scope scope
```

7. Set the server LDAP version; select 2 or 3; default 3.

```
sonic(config) # ldap-server version version
```

8. Set the LDAP server base value.

```
sonic(config) # ldap-server base string
```

9. Set the LDAP server SSL; select on, off, or start\_tls; default off.

```
sonic(config) # ldap-server ssl ssl
```

10. Set the LDAP server distinguished name to bind to.

```
sonic(config) # ldap-server binddn string
```

11. Set the LDAP server credentials to bind to.

```
sonic(config) # ldap-server bindpw string
```

12. Set the LDAP server PAM filter name.

```
sonic(config) # ldap-server pam-filter string
```

13. Set the LDAP server PAM group distinguished name (default uid).

```
sonic(config) # ldap-server pam-login string
```

14. Set the LDAP server PAM group attribute value.

```
sonic(config) # ldap-server pam-group string
```

15. Set the LDAP server PAM member attribute value.

```
sonic(config) # ldap-server pam-member string
```

16. Set the LDAP server sudo base distinguished name.

```
sonic(config) # ldap-server sudoers-base string
```

17. Set the LDAP server NSS search base.

```
sonic(config) # ldap-server nss-base string
```

18. Set the LDAP server NSS search base password.

```
sonic(config)# ldap-server nss-base-passwd string
```

19. Set the LDAP server NSS search base group.

```
sonic(config)# ldap-server nss-base-group string
```

20. Set the LDAP server NSS search base for shadow map.

```
sonic(config)# ldap-server nss-base-shadow string
```

21. Set the LDAP server NSS search base for netgroup map.

```
sonic(config)# ldap-server nss-base-netgroup string
```

22. Set the LDAP server NSS search base for sudoers map.

```
sonic(config)# ldap-server nss-base-sudoers string
```

23. Set the LDAP server NSS initialization groups ignore users value.

```
sonic(config)# ldap-server nss-initgroups-ignoreusers string
```

## Configuration

```
sonic(config)# ldap-server timelimit 13
sonic(config)# ldap-server bind-timelimit 10
sonic(config)# ldap-server idle-timelimit 12
sonic(config)# ldap-server retry 8
sonic(config)# ldap-server port 81
sonic(config)# ldap-server scope sub
sonic(config)# ldap-server version 2
sonic(config)# ldap-server base basetest
sonic(config)# ldap-server ssl on
sonic(config)# ldap-server binddn dnname
sonic(config)# ldap-server bindpw testpasswd
sonic(config)# ldap-server pam-filter testfilter
sonic(config)# ldap-server pam-login-attribute loginattrstring
sonic(config)# ldap-server pam-group-dn grpdn
sonic(config)# ldap-server pam-member-attribute attrstring
sonic(config)# ldap-server sudoers-base dnqrystr
sonic(config)# ldap-server nss-base-passwd dnsearchstr
sonic(config)# ldap-server nss-base-group grpmap
sonic(config)# ldap-server nss-base-shadow grpmap
sonic(config)# ldap-server nss-base-netgroup netgrpstr
sonic(config)# ldap-server nss-base-sudoers sudomap
sonic(config)# ldap-server nss-initgroups-ignoreusers grpstr
sonic(config)# no ldap-server nss-initgroups-ignoreusers grpstr
```

# Network Time Protocol

The Network Time Protocol (NTP) synchronizes the clocks in network devices. NTP coordinates time distribution in a large network between time servers and client devices. NTP clients synchronize with NTP servers to receive accurate time updates. NTP clients can choose from several NTP servers to determine which offers the best available source of time and the most reliable transmission of information.

You can set a switch to poll multiple NTP time-serving hosts. From these time servers, the switch chooses one NTP server to synchronize with, and acts as a client to the NTP server. NTP authentication allows a switch to validate NTP servers. After the host-client relationship establishes, the server routes time information through the network to the switch.

As an NTP client, the switch sends messages to one or more servers and processes the replies as received. Information in an NTP message allows each client/server peer to determine the timekeeping characteristics of its other peers, including the expected accuracies of their clocks. Using this information, each peer selects the best time from several other clocks, updates the local clock, and estimates its accuracy.

A switch can simultaneously act as an NTP client to a remote NTP server, and as an NTP server to downstream NTP clients, such as servers. Each NTP client determines if it uses NTP authentication to validate an upstream NTP server.

**i** **NOTE:** The NTP service on the switch can operate with or without NTP authentication for either a remote NTP server or a downstream NTP client.

**i** **NOTE:** When you configure certain NTP commands, the NTP process may restart.

### NTP defaults

- NTP server — Not configured
- NTP source interface — Not configured
- NTP in VRF instance — Default VRF
- NTP authentication key — Not configured
- NTP trusted key — Not configured
- NTP authentication — Disabled
- NTP prefer flag — False

## NTP configuration

To configure NTP on a switch:

1. Configure an NTP server by entering its IP address or domain name to synchronize time on the switch. You can configure multiple NTP servers using the `ntp server` command. (Optional) To use NTP authentication, enter an authentication-key ID (1 to 65535). This authentication key is used either by the switch to validate a remote NTP server as a time source, or by a downstream NTP client to validate the switch as a time source.

```
sonic(config)# ntp server {ipv4-address | ipv6-address | ntp-server-name} [key key-id]
[prefer true | false] [maxpoll interval] [minpoll interval]
```

To unconfigure an NTP server, enter the `no` version of the complete command without the authentication key ID.

2. (Optional) Configure NTP to operate either in the Management or default VRF. By default, NTP is enabled in the Management VRF if it is configured. If no Management VRF is configured, NTP service is enabled in the default VRF.

```
sonic(config)# ntp vrf {mgmt | default}
```

3. Configure the switch interface whose IPv4 or IPv6 address is used as the source address in packets sent to an NTP server. You can configure multiple global NTP source interfaces. If the switch serves as a time source, configure an interface to communicate with downstream NTP clients.

```
sonic(config)# ntp source-interface {interface-type interface-number}
```

Where `interface-type interface-number` is one of these values:

- `Eth slot/port[/breakout-port]`
- `PortChannel portchannel-number`
- `Vlan vlan-id`
- `Loopback number`
- `Management 0`

If no source interface is configured, by default a single NTP source interface is selected using an internal algorithm in the following order:

- a. Statically configured management interface IP address
- b. IP address configured on the loopback0 interface
- c. DHCP-acquired management IP address

**i** **NOTE:** If the management interface is DHCP enabled and the loopback0 interface has an IP address assigned, the system uses the IP address. In such a scenario, you must manually configure the source interface.

To remove an NTP source interface, enter the `no` version of the complete command.

4. (Optional) Configure the switch to authenticate a remote NTP server which serves as the time source to synchronize the local time. Using the authentication key, the switch rejects an NTP server if the received NTP packets do not pass the authentication check. NTP authentication is disabled by default. Enter the same commands to configure the switch as an NTP server that a downstream NTP client validates as an acceptable time source.

- a. Create an authentication key on the switch. Re-enter the command to create additional keys.
  - `key-id` defines the authentication-key number (1 to 65535; no default).

- The supported authentication types are md5, sha1 and sha2-256.
- Enter the authentication password in plain text the first time. The password is encrypted in the running configuration. In future authentication-key configuration, you can copy and paste the encrypted password (with the encrypted keyword) from the show running configuration output.

```
sonic(config)# ntp authentication-key key-id type password
```

- b. Configure the trusted authentication-key numbers (1 to 65535; created in Step 1) that the switch must receive in NTP packets in order to accept the NTP server time. Trusted keys identify trusted sources — the NTP servers from which the switch accepts time synchronization.

```
sonic(config)# ntp trusted-key id-number
```

- c. Enable NTP authentication on the switch.

```
sonic(config)# ntp authenticate
```

To disable NTP authentication, enter no ntp authenticate.

- d. Configure the same NTP authentication settings on a remote NTP server that serves as an NTP time source or on a downstream NTP client.

### DHCP-based NTP server configuration

The management IP address can be configured statically or acquired from a DHCP server. If the Management IP address is acquired using DHCP, and if the NTP server option on the DHCP server specifies an NTP server, the NTP servers that are acquired from the DHCP server take precedence over user-configured NTP servers on the switch.

The NTP source interface and the NTP VRF that you configure take effect only on user-configured NTP servers.

## NTP server preference configuration

NTP server preference configuration allows you to assign an NTP client or peer as the preferred time server. If you assign more than one preferred server, the order in which the NTP servers are used is the order in which you configured them. If the first configured server becomes unavailable, the second configured server is used, and so on. Commands are sent via Ntpq as stated in the restart avoidance design.

**i | NOTE:** In case of a system reboot or NTP restart, the order of preferred server configuration is stored in the NTP configuration file /etc/ntp.conf, and the server associations are recreated.

### Add NTP server as preferred server

To configure a server as the preferred NTP server, use the prefer keyword followed by true:

```
sonic(config)# ntp server 99.1.1.1 prefer true
```

### Remove the preferred NTP server configuration

To remove the preferred server configuration, use the prefer keyword followed by false:

```
sonic(config)# ntp server 99.1.1.1 prefer false
```

## Poll configuration

Configure the maximum and minimum intervals for polling a server. The default values for minpoll is 6, and maxpoll is 10.

```
sonic(config)# ntp server 134.214.100.6 minpoll 5 maxpoll 9
```

## Example: Configure switch as NTP client and NTP server

To configure a switch as an NTP client that uses NTP authentication and the Management interface to receive NTP packets from a remote NTP server:

```
! Configure switch as NTP client
sonic(config)# ntp server 100.94.121.15 key 1
sonic(config)# ntp source-interface Management0
sonic(config)# ntp authentication-key 1 md5 ntpclient2
sonic(config)# ntp trusted-key 1
sonic(config)# ntp authenticate
```

You can also configure the switch as an NTP server that serves as a time source to a downstream NTP client. As an NTP server, the switch supports NTP authentication of itself as a valid time source by the NTP client. In the following example, the NTP client communicates with the NTP server using Loopback 100, which has IPv6 address 2001:aa:aa::1/128.

```
! Configure switch as NTP server
sonic(config)# ntp source-interface Loopback 100
sonic(config)# ntp authentication-key 2 md5 jungle
sonic(config)# ntp trusted-key 2
sonic(config)# ntp authenticate

! Configure downstream server as NTP client
sonic(config)# ntp server 2001:aa:aa::1 key 2
sonic(config)# ntp source-interface Vlan 100
sonic(config)# ntp authentication-key 2 md5 jungle
sonic(config)# ntp trusted-key 2
sonic(config)# ntp authenticate
sonic(config)# ntp server 134.214.100.6 prefer true
sonic(config)# ntp server 134.214.100.6 key 1 prefer true
sonic(config)# ntp server 134.214.100.6 minpoll 5 maxpoll 9 prefer true
```

### View NTP configuration

If a switch acts as both an NTP client and NTP server:

```
sonic# show running-configuration
!
ntp authenticate
ntp authentication-key 1 md5 U2FsdGVkX1+PWrl46GIn1f6u476jGR842Ptc6I1t4BY= encrypted
ntp authentication-key 2 md5 U2FsdGVkX18cdzcWfBlp4/5EZkvlhs4OrA0wiQ4OT6o= encrypted
ntp server 100.94.121.15 key 1
ntp source-interface Management 0
ntp source-interface Loopback 100
ntp trusted-key 1
ntp trusted-key 2
!
```

On a downstream NTP client:

```
sonic# show running-configuration
!
ntp authenticate
ntp authentication-key 2 md5 U2FsdGVkX18nq6tgC9nau2EzaEJ7y8sRGfNN6LwhpY0= encrypted
ntp server 2001:aa:aa::1 key 2
ntp source-interface Loopback Vlan100
ntp trusted-key 2
ntp server 134.214.100.6 prefer true
ntp server 134.214.100.6 key 1 prefer true
ntp server 134.214.100.6 minpoll 5 maxpoll 9 prefer true
!
```

### View NTP servers

**(i) NOTE:** If you configure an NTP server using an NTP pool name, such as pool.ntp.org, and if the server cannot resolve the specified name, the NTP server is displayed in show ntp server output, but not displayed in show ntp associations.

```
sonic# show ntp server

NTP Servers minpoll maxpoll Prefer Authentication key ID

10.89.192.129 6 10 False
134.214.100.6 5 9 True
144.126.242.176 6 10 True
```

### View NTP global configuration

```
sonic# show ntp global

NTP Global Configuration

NTP source-interface: Management0
 Loopback100
NTP vrf: mgmt
```

### View NTP associations

Communications between a switch running an NTP client and an NTP server are called *associations*. Use the show ntp associations command to display the configured time servers and the reliability of their communication.

```
sonic# show ntp associations
 remote refid st t when poll reach delay offset jitter
=====
+10.11.0.1 10.11.8.1 4 u 9 64 1 0.232 -2.570 0.062
*10.11.0.2 10.11.8.1 4 u 8 64 1 0.262 -0.747 0.033
* master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

\* The switch is synchronized with this server or peer.

**number** The switch is not yet synchronized with this server or peer.

**+** The switch selected this server or peer for time synchronization.

**-** The switch considers this server or peer as a candidate for time synchronization.

**~** The server or peer is statically configured.

**remote** IP address of NTP server or peer.

**refid** IP address of the remote device with which the server or peer synchronizes.

**st** Stratum level — Number of hops that the NTP server or peer is from the time-source device, from 0 to 16. 0 indicates that the device is the time source. 1 indicates that the device is directly connected to the time source. 2 indicates that the device is connected to the stratum 1 device, and so on. 16 means that the device is not synchronized with a time source. As an NTP client, the switch automatically uses the server or peer with the lowest stratum number.

**t** Type of NTP device:

- u — Unicast or anycast NTP client
- b — Broadcast or multicast NTP client
- l — Local reference clock on switch
- s — Symmetric peer
- A — Anycast NTP server
- B — Broadcast NTP server
- M — Multicast NTP server

**when** Time (in seconds) since an NTP packet update was received or since the time was last synchronized.

**poll** Polling interval (in seconds) used by the switch to send NTP time requests, from 8 to 5160 (36 hours).

|               |                                                                                                                                                                                                                                                                      |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>reach</b>  | Reachability of NTP server — If the reach value is nonzero, the server is reachable; if the value is 0, the server is unreachable. The <code>reach</code> value is a peer variable that records when a valid NTP packet is received, and when an NTP packet is sent. |
| <b>delay</b>  | Round-trip delay (in milliseconds) to the NTP server.                                                                                                                                                                                                                |
| <b>offset</b> | Time difference (in milliseconds) between the switch and the NTP server or another NTP peer.                                                                                                                                                                         |
| <b>jitter</b> | Mean deviation in times between the switch and the NTP server based on multiple time samples.                                                                                                                                                                        |

## NTP troubleshooting

- If you enter multiple NTP configuration commands in quick succession, an internal timer lessens the need for numerous NTP service restarts. There may be a delay of three seconds for the complete set of NTP commands to take effect.
- When you configure an NTP server the first time, the system time may be updated by a large time difference. This large time interval can invalidate the authentication token used by the switch. As a result, an error message, such as `%Error: Invalid JWT Token`, displays after you enter CLI commands.

```
sonic(config)# ntp server 10.14.1.97
sonic(config)# do show ntp associations
%Error: Invalid JWT Token

sonic(config)# do show clock
%Error: Invalid JWT Token
```

To avoid this behavior, Dell Technologies recommends that after you configure a new NTP server, start a new CLI session.

```
sonic(config)# end
admin@sonic:~$

admin@sonic:~$ sonic-cli

sonic# show ntp associations
 remote refid st t when poll reach delay offset jitter
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
*10.14.1.97 10.14.1.38 4 u 67 64 17 4.245 -2.631 23.047
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
* master (synced), # master (unsynced), + selected, - candidate, ~ configured

sonic# show clock
Tue, 10 Nov 2020 10:52:11 UTC
sonic#
```

## DNS server

A domain name system (DNS) server translates hostnames to IP addresses. You can enter an easy-to-remember hostname, such as `www.dell.com`, in a command syntax instead of entering the complete IP address of the device. A configured DNS server looks up the IP address of the network device that is associated with the hostname.

The DNS service is not enabled by default. You must configure a source interface and one or more DNS servers from which the switch receives IP addresses.

1. Configure the source interface on the switch that is used for the source IP address in DNS queries.

```
sonic(config)# ip name-server source-interface {Eth slot/port[/breakout-port] | Loopback number | Management 0 | PortChannel number | Vlan vlan-id}
```

2. Configure a DNS server. Enter an IPv4 address in `A.B.C.D` format. Enter an IPv6 address in `A::B` format. Enter a VRF name to specify the VRF to use to reach the name server.

```
sonic(config)# ip name-server {ip-address | ipv6-address} [vrf vrf-name]
```

3. Reenter the `ip name-server` command to configure additional DNS servers.

- Verify the DNS server configuration.

```
sonic# show hosts
```

#### DNS server configuration

```
sonic(config)# ip name-server source-interface Loopback 0
sonic(config)# ip name-server 1.2.3.4 vrf mgmt
sonic(config)# ip name-server 2001:4860:4860::8888 vrf mgmt
sonic(config)# ip name-server 8.8.8.8
```

```
sonic# show hosts
Source Interface: Loopback0
Name servers are: 1.2.3.4(vrf: mgmt), 2001:4860:4860::8888(vrf: mgmt), 8.8.8.8
```

## Fast reboot

The fast reboot feature allows a switch to reboot quickly, with minimum disruption to the data plane and packet forwarding. Use a fast reboot to minimize traffic impact in data center networking operation during infrastructure maintenance and upgrade.

**i NOTE:** For a fast reboot to be effective, certain critical SONiC services must be running on the switch. To verify that these services are running, use the `show system status` SONiC CLI command (see [View core service status](#)).

### Run a fast reboot

In EXEC mode, enter the following Management Framework CLI command:

```
sonic# fast-reboot
```

## Simple Network Management Protocol

Network management stations use the Simple Network Management Protocol (SNMP) to retrieve and modify software configurations for managed objects on a local agent in network devices. A *managed object* is data of management information. Use SNMP data collection to monitor and manage switch performance, and troubleshoot error conditions.

The SNMP agent in a managed switch maintains the data for managed objects in management information bases (MIBs). Managed objects are identified by their object identifiers (OIDs). A remote SNMP agent performs an SNMP walk on the OIDs stored in MIBs on the local switch to view and retrieve information.

Enterprise SONiC supports standards MIBs, including all get requests. Enterprise SONiC does not support SNMP SET operations. MIBs are hierarchically structured and use object identifiers to access managed objects. For a list of MIBs supported in the Enterprise SONiC version running on a switch, see [MIBs](#).

Enterprise SONiC supports different SNMP versions for communication between SNMP managers and agents. SNMP versions provide different levels of security, such as user authentication and message encryption, in SNMP messages.

The local SNMP agent sends notifications of system events to configured management stations called *hosts*. SNMP notifications are sent for events, such as system reloads and loss of connection with neighboring devices. SNMP notifications can be traps or informs.

- An SNMP trap is sent when a change of state is detected in a management object. No acknowledgment is required from a management station that receives the trap message.
- An SNMP inform sends the trap content and requests confirmation of receipt from a management station. The inform is resent if no response is received. A management station sends its response as a protocol data unit (PDU).

## SNMP versions

Enterprise SONiC supports SNMPv2c and SNMPv3. You can use SNMPv2c and SNMPv3 at the same time for data collection and switch management. Each SNMP version provides different security levels:

- SNMPv2c uses community strings to authenticate SNMP management stations. SNMP messages are sent without encryption in plain text.
- SNMPv3 provides user-configured levels of security to authenticate users and encrypt SNMP messages:
  - no authentication — No user password or message encryption
  - authentication — User-password authentication only
  - privacy — User-password authentication and message encryption

### **SNMPv2c community strings**

SNMPv2c uses community strings, which function as passwords to allow user access to a managed switch. Community strings are sent with Get requests to retrieve information from the local SNMP agent on a switch.

To configure a community string for SNMPv2c, use the `snmp-server community` command.

### **SNMPv3 security**

SNMPv3 provides an enhanced security model for user authentication and SNMP message encryption. User authentication requires that SNMP packets come from an authorized source. Message encryption ensures that packet contents cannot be viewed by an unauthorized source.

To configure SNMPv3-specific security settings — user authentication and message encryption — use the `snmp-server user` command.

## **MIBs**

Enterprise SONiC supports the following MIBs. For information about the MIB object OIDs supported by Enterprise SONiC, see [MIB objects](#).

**Table 11. Supported MIBs**

| Supported MIBs                 |                         |
|--------------------------------|-------------------------|
| BGP4-MIB                       | IP-MIB                  |
| BRCM-PROD-MIB                  | LLDP-MIB                |
| BRIDGE-MIB                     | NET-SNMP-AGENT-MIB      |
| BROADCOM-SONIC-CONF-MGMT-MIB   | NET-SNMP-MIB            |
| BROADCOM-SONIC-PRODUCT-MIB     | NOTIFICATION-LOG-MIB    |
| cisco.ciscoEntityFruControlMIB | OSPF-MIB                |
| cisco.ciscoPfcExtMIB           | Q-BRIDGE-MIB            |
| cisco.ciscoSwitchQosMIB        | RFC1213-MIB             |
| Dell-VENDOR-MIB.mib            | SNMP-FRAMEWORK-MIB      |
| DISMAN-EVENT-MIB               | SNMP-MPD-MIB            |
| ENTITY-MIB                     | SNMP-TARGET-MIB         |
| ENTITY-SENSOR-MIB              | SNMP-USER-BASED-SM-MIB  |
| ENTITY-STATE-MIB               | SNMP-VIEW-BASED-ACM-MIB |
| HOST-RESOURCES-MIB             | SNMPv2-MIB              |
| IF-MIB                         | TCP-MIB                 |
| IP-FORWARD-MIB                 | UDP-MIB                 |
|                                | UCD-SNMP-MIB            |

## **SNMP traps**

Enterprise SONiC supports the following SNMP traps:

- authenticationFailure
- bgpBackwardTransNotification

- bgpEstablishedNotification
- coldStart
- configChangeTrap
- linkup/linkDown
- ospfNbrStateChange
- ospfIfStateChange
- warmStart

## Configure SNMP

To set up communication between the local SNMP agent on the switch and remote management stations in the network, perform these tasks in any order. By default, no SNMP settings are configured.

For SNMPv2c:

- Configure community strings to communicate with management stations — see [Configure SNMP communities](#).
- (Optional) Configure views of the MIB tree structure — see [Configure SNMP views](#).
- (Optional) Configure SNMP groups — see [Configure SNMP groups](#).
- Configure user access to the SNMP agent on the switch — see [Configure SNMP users](#).
- Enable SNMP traps and informs — see [Configure SNMP notifications](#).
- (Optional) Configure the local agent address — see [Configure local agent address](#).

For SNMPv3:

1. Configure views of the MIB tree structure — see [Configure SNMP views](#).
2. Configure SNMP groups — see [Configure SNMP groups](#).
3. Configure user access with authentication and encryption — see [Configure SNMP users](#).
4. Enable SNMP traps and informs — see [Configure SNMP notifications](#).
5. Configure the SNMP engine ID — see [Configure SNMP engine ID](#).
6. (Optional) Configure the local agent address — see [Configure local agent address](#).

(Optional) For SNMP troubleshooting, configure an SNMP contact and location — see [Configure SNMP contact and location](#).

## Configure SNMP communities

Use a community string to authenticate users in SNMPv2c communication with management stations. A community string serves as a password that is included in Get requests to allow user access to a managed switch, and that allows the switch to send SNMP messages to an authenticated user.

In SNMPv2c, a community string supports all alphanumeric and special characters except space, comma, and @; 32 characters maximum. A minimum of four characters is required. In addition, using # as the first character in a community string (for example, `snmp-server community #public`) is not supported.

For example, the community string `#test123` is not valid while the community string `test#123` is supported:

```
sonic(config)# snmp-server community #test123
% Error: Invalid input detected at "^" marker.

sonic(config)# snmp-server community test#123
```

Specify a group name to use a community string to authenticate an individual or group of users; 32 characters maximum. To set up a user group, see [Configure SNMP groups](#).

```
sonic(config)# snmp-server community community-string [group group-name]
```

To remove a community string, enter the `no` version of the command.

### Configure SNMP community

```
sonic(config)# snmp-server community comm1 group group-lab
```

## Display SNMP communities

```
sonic# show snmp-server community
Community Name Group Name
----- -----
comm1 group-lab
comm2 None
```

## Configure SNMP views

Configure one or more views of the MIB tree structure allowed for Get and Set requests from a management station. Enter an *oid-tree* value to specify the OID in the MIB tree hierarchy at which a view starts. Enter included or excluded to include or exclude the rest of the sub-tree MIB contents in the view. If necessary, re-enter the command to exclude tree entries in the included content. By default, no SNMP views are configured.

```
sonic(config)# snmp-server view view-name {oid-tree [included | excluded]}
```

To remove a configured view, enter the `no snmp-server view view-name oid-tree` command.

### Configure SNMP view

```
sonic(config)# snmp-server view view2 1.2.3.4.5.6.7.8.9.2 excluded
```

## Display SNMP views

```
sonic# show snmp-server view
View Name OID Tree Type
----- ----- -----
view1 1.2.3.4.5.6.7.8.9.1 included
view2 1.2.3.4.5.6.7.8.9.5.1 excluded
```

## Configure SNMP groups

Configure an SNMP access group with the views allowed for the members of the group. Specify the read-only, read/write, and/or notification views; 32 characters maximum.

Enter v2c or v3 for the SNMP version used to send and receive Get and Set requests. Enter any to use both SNMPv2c and SNMPv3 on a switch.

SNMPv2c uses community strings to authenticate user groups — see [Configure SNMP communities](#). For SNMPv3, specify the level of security to use for user authentication and privacy settings:

- noauth — Do not authenticate users or encrypt SNMP messages; send messages in plain text.
- auth — Authenticates users in SNMP messages.
- priv — Authenticates users and encrypts/decrypts SNMP messages.

To configure the authentication and privacy settings for individual SNMPv3 users, use the `snmp-server user` command; see [Configure SNMP users](#).

Specify the views allowed for a user group. By default, no views are configured. To configure a view of the MIB tree on the switch, see [Configure SNMP views](#).

```
sonic(config)# snmp-server group group-name {any | v2c | v3 {auth | noauth | priv}}
[read view-name] [write view-name] [notify view-name]
```

To remove an SNMP group, enter the `no snmp-server group group-name` command.

### Configure SNMPv2c user group

```
sonic(config)# snmp-server group group1 v2c notify no_view
```

## Configure SNMPv3 user group

```
sonic(config)# snmp-server group group-floor2 v3 priv read r_view write w_view notify n view
```

## Display SNMP groups

```
sonic# show snmp-server group
```

| Group        | Name | Model:            | Security | Read View | Write View | Notify View |
|--------------|------|-------------------|----------|-----------|------------|-------------|
| group-floor1 | v2c  | no-auth-no-priv   | ro_view  | wr_view   | None       |             |
| group-floor2 | v3   | : auth-priv       | r_view   | None      | None       |             |
| group-lab    | v2c  | : no-auth-no-priv | None     | None      | None       |             |

## Configure SNMP users

Configure remote SNMPv3 user access to the local agent on the switch. (Optional) Assign each user to a group membership and configure SNMPv3-specific authentication and encryption settings. Enter an authentication or privacy password in plain text or as an encrypted string. By default, no passwords are configured. Use the `encrypted` option to enter already encrypted authentication and privacy passwords. Re-enter the command multiple times to configure SNMP security settings for additional users.

```
sonic(config)# snmp-server user user-name [group group-name] [encrypted] [auth {md5 | sha | noauth} [auth-password password]] [priv {DES | AES-128} [priv-password password]]
```

- `user user-name` — Name of remote SNMPv3 user that connects to the local agent; 32 characters maximum.
  - `group group-name` — SNMP group to which the user is assigned; 32 characters maximum. The SNMP group determines a user's access privilege: read, read/write, or notify. To configure a group's access privilege, use the `snmp-server group` command; see [Configure SNMP groups](#).
  - `encrypted` — For SNMPv3, specifies that a user's password is sent in encrypted format.
  - `auth {md5 | sha | noauth}` — For SNMPv3, specifies the authentication protocol used: MD5 or SHA. For SNMPv2c, enter noauth for no authentication.
  - `auth-password password` — For SNMPv3, specifies the password used to authenticate a remote user; 64 characters maximum.
  - `priv {DES | AES-128} priv-password password` — For SNMPv3, specifies the encryption algorithm and user privacy password; 64 characters maximum.

SNMPv3 provides the strongest security with user authentication and packet encryption. No default values exist for SNMPv3 authentication and privacy passwords. If you forget a password, you cannot recover it — you must reconfigure the user. You can specify either a plain-text password or an encrypted cypher-text password. In either case, the password stores in the configuration in encrypted form and displays as encrypted in the `show running-config snmp` output.

**i** **NOTE:** An SNMP user configuration is tied to the engine ID of the switch — see [Configure SNMP engine ID](#). If you copy an SNMP user configuration from one switch to another, an SNMP query will fail.

## Configure SNMPv3 users

```
sonic(config)# snmp-server user user1 group group-lab auth md5 auth-password pwd priv aes-128 priv-password pwd
```

To remove an SNMPv3 user, enter the `no snmp-server user user-name` command.

## Display SNMPv3 users

```
sonic# show snmp-server user
```

| User Name | Group Name | Auth | Privacy |
|-----------|------------|------|---------|
|           |            |      |         |

|       |              |      |         |
|-------|--------------|------|---------|
| user1 | group-lab    | md5  | aes-128 |
| user2 | group-floor2 | None | None    |

## Configure SNMP notifications

An SNMP agent sends notification of events to a management station using:

- SNMP traps, which are unsolicited SNMP messages. SNMP traps optimize the use of network resources.
- SNMP informs, which require an acknowledgment from a network management station. Informs are more reliable than traps. If an SNMP agent does not receive an acknowledgment, it resends the inform with a maximum of three retries.

Configure a management station which receives notifications using the `snmp-server user` command. Enter a station's IP or IPv6 address. Then enable the switch to send SNMP traps and informs.

- For SNMPv2c, specify the IP address and the SNMP community name that is used to control remote access to the local agent — see [Configure SNMP communities](#). An SNMP community name acts as a password to allow only configured host addresses to receive SNMP notifications. A community string supports all alphanumeric and special characters except space, comma, and @; 32 characters maximum. A minimum of four characters is required. In addition, using # as the first character in a community string (for example, `snmp-server host 1.2.3.4 community #public`) is not supported.
- For SNMPv3, specify the IP address and username configured with authentication and encryption settings — see [Configure SNMP users](#).

To send SNMP notifications as traps, enter `traps` and a security level. To send SNMP notifications as informs, enter `informs` and a security level. By default traps and informs are sent on UDP port 162 and the default VRF.

For `timeout`, enter the number of seconds before an idle session times out. For `retries`, enter the number of times that the agent re-sends an SNMP notification after timing out.

- Configure remote management stations to receive SNMPv2c notifications and access the local agent.

```
sonic(config)# snmp-server host {ipv4-address | ipv6-address} community community-name {traps v2c | informs [timeout seconds] [retries number]} [source-interface {Eth slot/port[/breakout-port] | Vrf vrf-name} [port udp-port-number]]
```

To remove a remote management station from receiving SNMPv2 notifications, enter the `no snmp-server host {ipv4-address | ipv6-address} community community-name` command.

- Configure remote management stations to receive SNMPv3 notifications and access the local agent.

```
sonic(config)# snmp-server host {ipv4-address | ipv6-address} user username {traps {auth | noauth | priv} | [informs {auth | noauth | priv} [timeout seconds] [retries number]]} [source-interface {Eth slot/port[/breakout-port] | Vrf vrf-name} [port udp-port-number]]
```

To remove a remote management station from receiving SNMPv3 notifications, enter the `no snmp-server host {ipv4-address | ipv6-address} user user-name` command.

- Enable all SNMP traps and informs that are generated on switch interfaces to be sent to an SNMP management station from the local agent.

```
sonic(config)# snmp-server enable trap
```

To disable the sending of SNMP traps and informs, enter the `no snmp-server enable trap` command.

- (i) NOTE:** By default, linkup and linkdown traps are enabled on all physical port interfaces. To disable or re-enable sending linkup and linkdown traps that are generated on an interface, use the `[no] snmp trap enable` command. For example:

```
sonic(config)# interface Eth1/2
sonic(config-if-Eth1/2)# no snmp trap enable

sonic(config)# interface Eth1/2
sonic(config-if-Eth1/2)# snmp trap enable
```

### Configure SNMPv2c trap receivers

```
sonic(config)# snmp-server host 1.2.3.4 community comm1 traps v2c
sonic(config)# snmp-server host 2.3.4.5 community public traps v2c port 1492 vrf Vrf1
```

```
sonic(config)# snmp-server host 2001::1 community comm2 informs timeout 150 retries 5
sonic(config)# snmp-server host 1.1.1.1 community abcd traps v2c source-interface
Ethernet 0
```

### Configure SNMPv3 trap receivers

```
sonic(config)# snmp-server host 1.2.3.5 user user1 informs noauth timeout 200 retries 10
sonic(config)# snmp-server host 3001::1 user user2 traps priv
```

### Display SNMPv2c and SNMPv3 trap receivers (hosts)

```
sonic# show snmp-server host

Target Address Port Type Community Ver T-Out Retries VRF Source-Interface
----- -----
100.94.58.239 162 trap fource v2c 15 3 Eth1/1
1001::1 inform user1 auth-priv v3 200 10 Eth1/2
```

### Display per-interface SNMP trap status

```
sonic# show snmp-server interface-traps

Interface Name Status
----- -----
Ethernet12 disable
```

## Configure SNMP engine ID

An engine ID identifies the SNMP local agent on a switch. The engine ID is an octet number; for example, 80000137031c721dc3c2e0. The local engine ID is used to set the local SNMP configuration. By default, the SNMP engine ID is derived from the MAC address.

```
sonic(config)# snmp-server engine engine-id
```

To remove the configured local engine ID, enter the no version of the command.

### Configure SNMP engine ID

```
sonic(config)# snmp-server engine 80:00:01:37:03:e8:b5:d0:cc:0f:cc
```

### Display SNMP engine ID

```
sonic# show snmp-server

Location : Bldg4 lab
Contact : Networking Support
EngineID : 80:00:01:37:03:e8:b5:d0:cc:0f:cc
Traps : enable
```

## Configure local agent address for Management VRF

To receive management station requests, the local agent listens by default on UDP port 161 and uses the default/Management VRF ([Management VRF](#)). However, you must configure the local SNMP agent to listen for requests from management stations on specified IP/IPv6 addresses, and (optionally) UDP ports or the Ethernet interface. User-defined nondefault VRFs are not supported. Re-enter the command to configure additional local agent addresses. The valid UDP port numbers are 1024 to 65535.

```
sonic(config)# snmp-server agentaddress ip-address [port udp-port-number] [interface vrf-name]
```

To remove a local agent address, enter the no version of the complete command.

## Configure local agent IP address for Management VRF

```
sonic(config)# interface Management 0
sonic(conf-if-Management0)# ip address 10.1.1.10/32
sonic(conf-if-Management0)# exit

sonic(config)# ip vrf mgmt

sonic(config)# snmp-server agentaddress 1.2.3.5 port 1024 interface Vrf1
sonic(config)# snmp-server agentaddress 1.2.3.5 port 1024 interface Eth1/10
```

## Display local agent addresses

```
sonic# show snmp-server
...
Agent Addresses:

IP Address UDP Port Interface
----- ----- -----
1.2.3.4 161
1.2.3.4 1024
1.2.3.5 1024 Eth1/10
```

## Configure SNMP contact and location

(Optional) For system troubleshooting, configure contact information (for example, phone number, email, tech support name) and the physical location (campus building, floor, room) of the local SNMP agent, use the `snmp-server engineID` command. Enter up to 32 characters for each string. Enclose each text in double quotes ("").

```
sonic(config)# snmp-server engineID [contact text] [location text]
```

To remove the configured SNMP contact or location, enter the `no snmp-server engineID contact` or `no snmp-server engineID location` command.

### Configure SNMP contact and location

```
sonic(config)# snmp-server contact "Dell Support"
sonic(config)# snmp-server location "Lab1, Rack-10"
```

### Display SNMP contact and location

```
sonic# show snmp-server

Location : Lab1, Rack-10
Contact : Dell Support
EngineID : 8000013703525400f6817e
Traps : enable
...
```

## Secure SNMP access

To secure the local SNMP agent from a denial of service (DoS) attack, apply a control-plane ACL that permits only the IP addresses of allowed remote management stations; for example:

```
sonic(conf-if-Eth1/2)# ip access-list ACL_IPV4_MGMT_ACCESS
sonic(conf-ipv4-acl)# seq 10 permit ip 100.0.24.0/24 any
sonic(conf-ipv4-acl)# seq 20 permit ip 172.16.24.0/24 any
sonic(conf-ipv4-acl)# seq 30 permit ip 100.0.9.0/26 any
sonic(conf-ipv4-acl)# seq 40 permit ip 100.10.1.32/27 any
sonic(conf-ipv4-acl)# seq 50 permit ip 100.252.5.64/28 any
sonic(conf-ipv4-acl)# seq 200 deny ip any any
sonic(conf-ipv4-acl)# exit
```

```
sonic(config)# line vty
sonic(config-line-vty)# ip access-group ACL_IPV4_MGMT_ACCESS in
```

## Example: Configure SNMPv2c

This example shows how to configure SNMPv2c on the switch, including community string and SNMP traps.

```
sonic(config)# snmp-server community dell
sonic(config)# snmp-server contact "Networking Support"
sonic(config)# snmp-server location "Bldg4 lab"
sonic(config)# snmp-server enable trap
sonic(config)# snmp-server host 172.17.100.81 community dell
sonic(config)# snmp-server host 172.17.100.81 community dell informs timeout 200 retries
10

sonic(config)# do show snmp-server
Location : Bldg4 lab
Contact : Networking Support
EngineID : 80000137031c721dc3c2e0
Traps : enable

sonic(config)# do show snmp-server host
Target Address Port Type Community Ver T-Out Retries
----- -----
172.17.100.81 162 trap dell v2c 15 3
```

## Example: Configure SNMPv3

This example shows how to configure SNMPv3 on the switch, including SNMP groups, users, MIB views, traps, and engine ID.

```
sonic(config)# snmp-server group systemtest v3 auth read rview
sonic(config)# snmp-server user delluser group systemtest auth md5 auth-password
dellPassword priv des priv-password dellPrivPassword
sonic(config)# snmp-server view rview .1 included
sonic(config)# snmp-server engine 800002de11f015ce10
sonic(config)# snmp-server contact "Networking Support"
sonic(config)# snmp-server location "Bldg4 lab"
sonic(config)# snmp-server enable trap
sonic(config)# snmp-server host 172.17.100.81 community dell
sonic(config)# snmp-server host 172.17.100.81 community dell informs timeout 200 retries
10

sonic(config)# do show snmp-server user
User Name Group Name Auth Privacy
----- ----- ----- -----
delluser systemtest md5 des

sonic(config)# do show snmp-server group
Group Name Model: Security Read View Write View Notify View
----- ----- ----- ----- -----
systemtest v3 : auth-no-priv rview None None

sonic(config)# do show snmp-server view
View Name OID Tree Type
----- ----- -----
rview .1 included

sonic(config)#do show snmp-server
Location : Bldg4 lab
Contact : Networking Support
EngineID : 800002de11f015ce10
Traps : enable
```

## Example: Configure agent address for Management VRF

For SNMPv2c and SNMPv3, you can use the Management VRF for out-of-band switch management. This example shows how to configure the local SNMP agent to listen for requests from management stations on a specified IP address over the Management VRF.

```
sonic(config)# interface Management 0
sonic(conf-if-Management0)# exit
sonic(config)# ip vrf mgmt
sonic(config)# snmp-server agentaddress 172.17.100.42 port 161 interface mgmt

sonic# show ip vrf
VRF-NAME INTERFACES

default
mgmt Management0

sonic# show ip interfaces
Flags: U-Unnumbered interface, A-Anycast IP

Interface IP address/mask VRF Admin/Oper Flags

Management0 172.17.100.42/24 mgmt up/up
```

### Configure SNMPv2 traps in Management VRF

To send SNMPv2c traps and informs over the Management VRF to a specified management station:

```
sonic(config)# snmp-server host 172.17.100.81 community dell traps v2c interface mgmt
sonic(config)# snmp-server host 172.17.100.81 community dell informs interface mgmt

sonic(config)# do show snmp counters
SNMP Counters
=====
39683 SNMP packets input
0 Bad SNMP version errors
12 Unknown community name
0 Illegal operation for community name supplied
0 Encoding errors
39671 Number of requested variables
0 Number of altered variables
2 Get-request PDUs
39669 Get-next PDUs
0 Set-request PDUs
39880 SNMP packets output
0 Too big errors (Maximum packet size 1500)
0 No such name errors
0 Bad values errors
0 General errors
39671 Response PDUs
209 Trap PDUs
```

 **NOTE:** The `show snmp counters` output is not available on an Edge bundle switch using the MF-CLI or a REST API request.

## Example: Read service tag using SNMP

To retrieve the service tag of an Enterprise SONiC switch for technical support or other reasons, you can use an `snmpget`, `snmpgetnext`, or `getbulk` operation. In the following examples, JB88PK2 is the service tag that is retrieved.

To configure the community string that is used a password in get requests, use the `snmp-server community` command — see [Configure SNMP communities](#).

```
sonic(config)# snmp-server community comm1
snmpwalk -v2c -c comm1 10.1.1.1 .1.3.6.1.4.1.674.10895.3000.1.2.100
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.1 = STRING: "Z9264F-ON"
```

```

SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.3 = STRING: "Dell EMC"
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.8.1.2.1 = STRING: "TW0WCXFVDNT0003H0065"
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.8.1.4.1 = STRING: "JB88PK2"
#
snmpgetnext -v2c -c comm1 10.1.1.1 .1.3.6.1.4.1.674.10895.3000.1.2.100.8.1.4
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.8.1.4.1 = STRING: "JB88PK2"
#
snmpget -v2c -c comm1 10.1.1.1 .1.3.6.1.4.1.674.10895.3000.1.2.100.8.1.4.1
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.8.1.4.1 = STRING: "JB88PK2"
#
snmpbulkget -v2c -c comm1 -r4 10.1.1.1 .1.3.6.1.4.1.674.10895.3000.1.2.100
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.1 = STRING: "Z9264F-ON"
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.3 = STRING: "Dell EMC"
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.8.1.2.1 = STRING: "TW0WCXFVDNT0003H0065"
SNMPv2-SMI::enterprises.674.10895.3000.1.2.100.8.1.4.1 = STRING: "JB88PK2"

```

## Dynamic Host Configuration Protocol

Dynamic Host Configuration Protocol (DHCP) is a network protocol that simplifies the assignment of the hostname, IP addresses, and other information to network devices. In addition to IP addresses, DHCP also assigns subnet mask, default gateway address, domain name server (DNS) address, and other configuration parameters.

**i** **NOTE:** The default hostname of a switch is `sonic`. You can configure a hostname using the command-line interface (`hostname` command) or allow DHCP to provide a hostname. The following precedence is used:

- The user-configured hostname always takes precedence over a DHCP-configured hostname.
- A DHCPv4-assigned hostname always takes priority over a DHCPv6-assigned hostname.
- A hostname that is received through DHCP is configured only if the current hostname is the default value `sonic`.
- The user-configured hostname `sonic` is not honored if a DHCP-assigned hostname exists.

## DHCP relay

Enterprise SONiC supports DHCP relay. DHCP relay is any device that forwards DHCP packets between DHCP clients and DHCP servers between different subnets. You can configure your switch to function as a DHCP relay in a network.

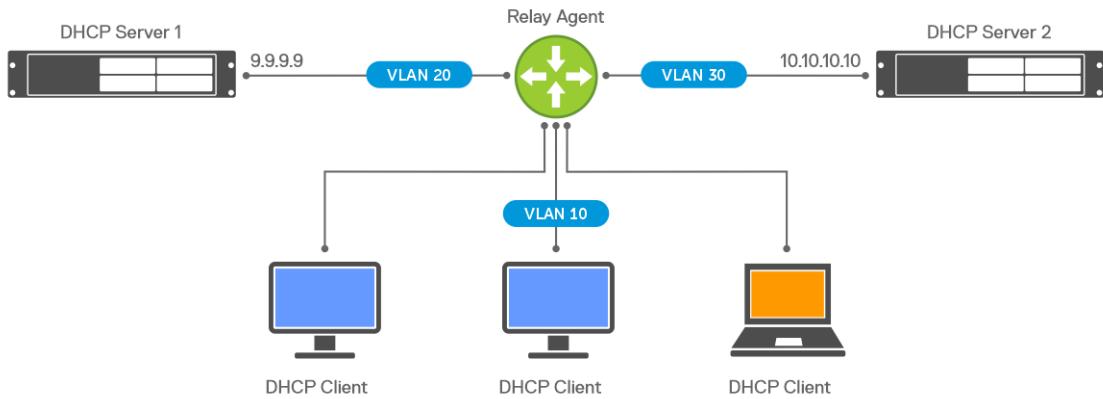
When a DHCP client requests for an IP address from a DHCP server, the client is not aware of the subnet it is going to belong. When a client is connected to a network, it sends a DHCP DISCOVER message as a broadcast. If the DHCP server resides within the same LAN or VLAN, the server assigns an IP address to the client directly.

If the DHCP server resides in a different broadcast domain, routers in the network do not forward the DHCP DISCOVER messages from clients by default. If you configure a device as a DHCP relay agent in your network, the relay agent can receive the DHCP DISCOVER broadcast messages, and send a unicast request to the DHCP server on behalf of the DHCP client.

A DHCP relay agent enables DHCP clients to receive IP addresses from a DHCP server, even if the server is in a different network or VLAN.

### DHCP relay operation

In the following figure, DHCP clients are connected to the interface of a relay device which belongs to VLAN 10. DHCP Server 1 and DHCP Server 2 are connected to VLAN 20 and VLAN 30 respectively. When you configure DHCP relay on the device, it forwards DHCP requests to the respective DHCP server and the reply from the server to clients.



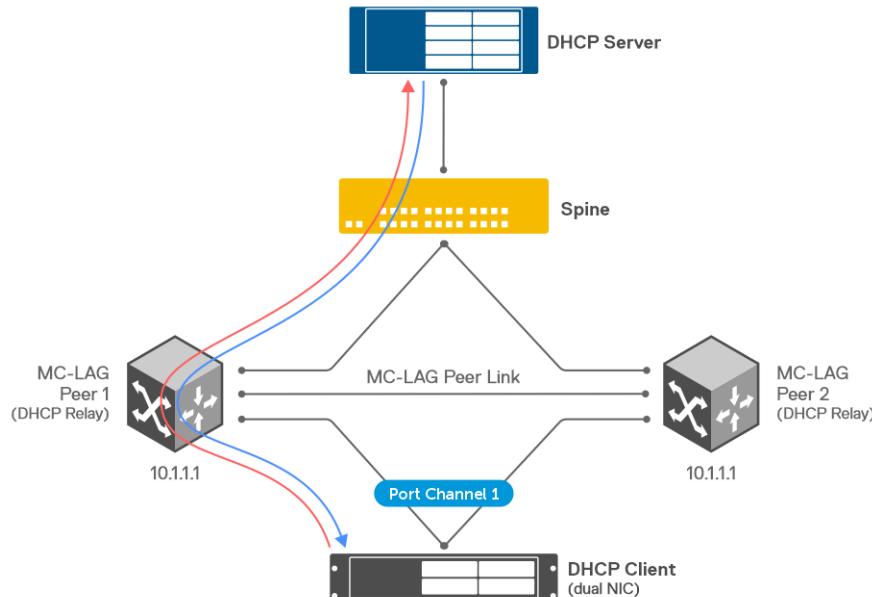
### DHCP relay in an MLAG setup

In an MC-LAG setup, a DHCP client sends a DHCP DISCOVER packet to either of the MLAG peers. When one of the peers receives the packet, it relays the packet to the DHCP server. When the DHCP server sends a response, the same MLAG peer that relayed the DHCP DISCOVER packet relays the reply from the server to the client.

### Configuration guidelines for MLAG

- Configure the same DHCP servers on both the MLAG peer switches.
- Configure DHCP relay to use link-selection and source interface options. This configuration ensures that the response from the server is received by the switch that relayed the DHCP packet.
- Ensure that the DHCP server is reachable from both MLAG peers.

In the following figure, a DHCP client sends a DHCP DISCOVER message. MLAG Peer 1 receives the message and relays it to the DHCP server through the Spine switch. The DHCP server sends a response to the same MLAG peer which, in turn, relays the information to the client.



### DHCP relay information option

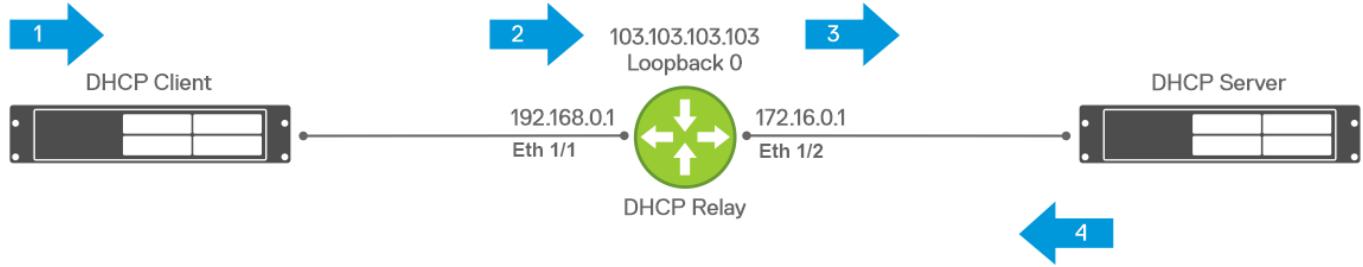
In a network where a single routing is involved, the DHCP relay uses the gateway IP address that is found in the giaddr field of the relayed packet to assign an IP address to the DHCP client. Rogue actors can spoof DHCP requests to gain unauthorized access to the network.

To prevent rogue devices from gaining access to the network, network administrators may place clients' DHCP servers in different networks. You can use the DHCP relay information option or DHCP option 82 to explicitly specify the subnet on which the DHCP client resides. The relay agent adds the suboption to the packet to specify the client subnet and the DHCP server uses the suboption value, instead of giaddr, to assign the DHCP address and lease.

**i | NOTE:** The link-selection suboption is applicable for DHCPv4 clients only and is not applicable for DHCPv6 clients.

The following figure illustrates how DHCP relay link selection works.

1. A DHCP client sends a DHCP request.
2. The relay agent receives the broadcast packet and adds the link-selection suboption with 192.168.0.1 address.
3. The relay agent sets the giaddr field that is based on the configured source interface. If source interface is configured as loopback 0, the giaddr is set to 103.103.103.103. The giaddr must be reachable from the server.
4. The DHCP server identifies the client subnet from the link-selection option and allocates address from the 192.168.0.x address pool. The server generates the offer packet and sends it to the IP address specified in the giaddr.



### Hop limit

DHCP clients usually set the hop count field in the DHCP packet to 0. When forwarding DHCP packets, the relay agent increments the hop count by 1. If the hop count in the DHCP packet is greater than or equal to the maximum number of hops configured on the relay agent, the device discards the packet.

The hop limit ensures that the DHCP packets are not looped in the network where multiple relay agents are present. The default value of the maximum number of hops is 10 which you can configure to a value from 1 to 16.

The hop limit configuration is per-interface (client facing) and applies to both DHCPv4 and DHCPv6 packets. The hop limit is only enforced for packets that are relayed to the server, it is not applicable for response packets that are received from a server. The interface counter is maintained to track the number of packets dropped due to the hop limit.

### Source interface selection

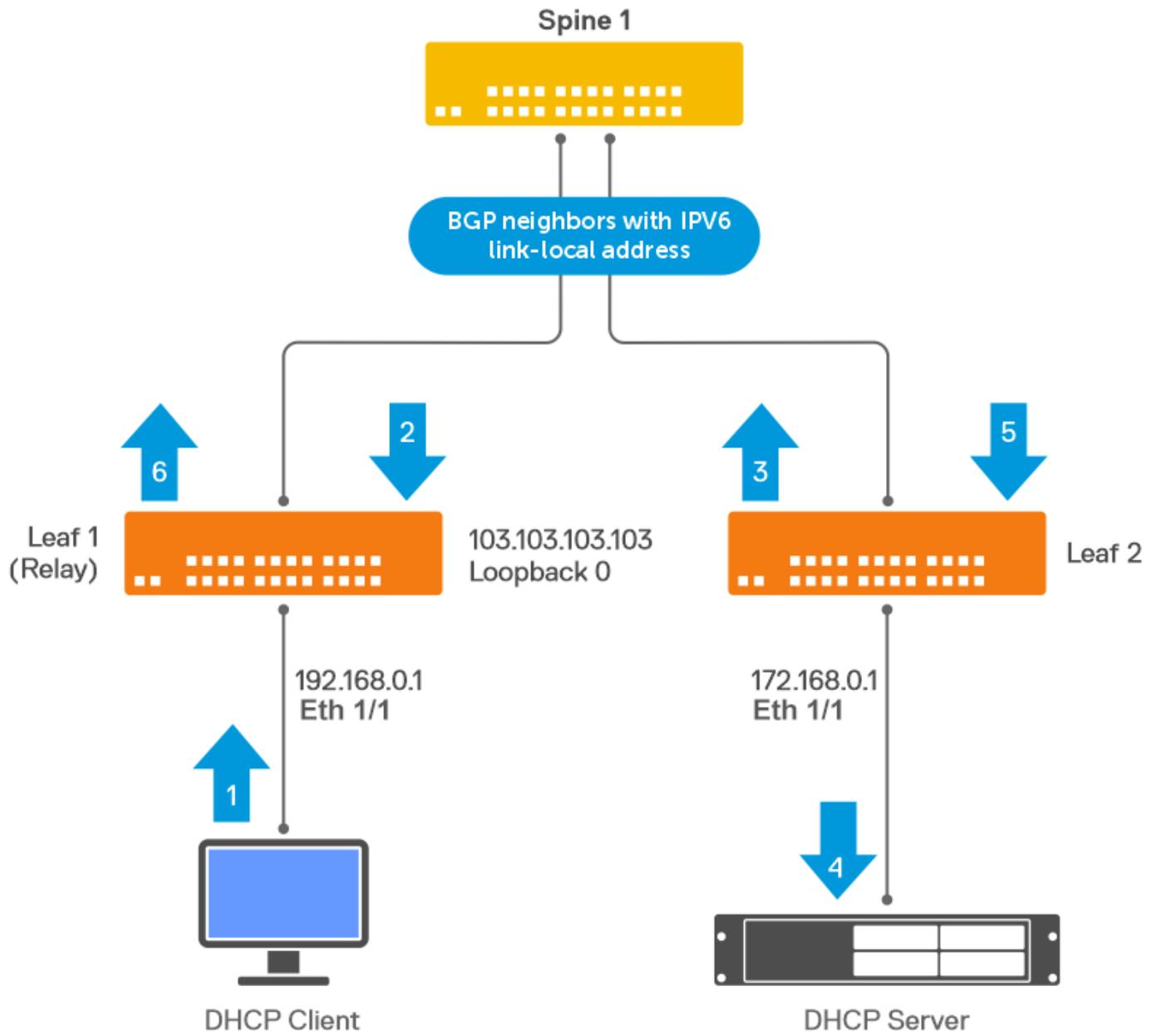
DHCP relay provides a source interface configuration option which specifies the source address to be used for relayed packets. If you do not specify the source interface, the source IP address in the relayed packet is automatically determined based on the outgoing interface. The system chooses the first address (IPv4 or IPv6) configured on the interface which falls in the same network as the destination address or next hop router.

The source interface configuration option is per-interface that is client facing and applies to both DHCPv4 and DHCPv6 packets. If the configured source interface does not have any IP address, the source IP address in the relayed packet is determined based on the outgoing interface. If you modify the address on the source interface, the relay agent uses the updated IP address for relaying packets.

If the link-selection suboption is enabled, configure a source interface that is reachable from the server. If you do not configure a source interface, the link-selection suboption is not added to the relayed packet.

### DHCP relay over IPv6 next hops

In data center network deployments as shown in the following figure, the DHCP server is reachable through an IPv6 underlay network. DHCP relay is enabled on the Leaf1 switch, which has BGP neighborship with Spine1. The DHCP server is connected to Leaf2 switch which also has BGP neighborship with Spine1.



The interfaces between the leaf and spine switches do not have IPv4 addresses, but they are enabled for IPv6 forwarding using link-local addresses. BGP peering between Leaf and Spine switch is established using IPv6 link-local addresses. BGP supports RFC 5549, which allows an IPv4 prefix to be carried over an IPv6 next hop.

On Leaf1, the IPv4 route to DHCP server is learned through BGP and points to the link-local next hop address of Spine1. Spine1 also has an IPv4 route that points to the link-local next hop address of Leaf2.

The following explains how DHCP relay works over IPv6 next hops:

1. DHCP client generates request.
2. The relay agent on Leaf1 is configured to use Loopback0 as the source interface. The relay agent sets the giaddr and source IPv4 address to 103.103.103.103, and forwards the request to the DHCP server whose IP address is 172.16.0.2 as per the BGP RFC 5549 route.
3. Leaf2 receives the relayed DHCP request from Spine1 and forwards it to the DHCP server which is directly connected.
4. The DHCP Server receives the relayed DHCP request, generates an offer packet, and sends it to the IP address specified in the giaddr, which is the Leaf1 loopback address 103.103.103.103.
5. Leaf2 has a BGP RFC5549 route to reach the loopback address of Leaf1 which is 103.103.103.130. The DHCP offer is forwarded to the relay agent as per the BGP route.
6. Leaf1 receives the response from DHCP server, strips option 82, and forwards it to the client.

## DHCP relay between VRFs

The DHCP relay agent supports forwarding of client requests to a server that is located in a different VRF. For example, the client is connected to an interface bound to the default VRF, and the server can reside in user VRF.

For such deployments, You can configure an option to specify the VRF name in which the DHCP server resides. If you do not specify the VRF name, the system assumes that the DHCP server resides in the default VRF. DHCP relay supports configuring multiple DHCP servers for a given client interface, and all these DHCP servers must reside in the same VRF. You can configure only one server VRF per client-facing interface.

**(i) NOTE:** The client VRF is derived from the interface on which relay is configured.

If you are using DHCPv4 relay, you must enable link-selection when the client and the server are in different VRFs. The link-selection suboption must use an interface that is bound to server VRF. This configuration is required to ensure that the response from the server is received by the DHCPv4 relay. If the client and server are in the same VRF, you need not configure link-selection as the giaddr is also in the server VRF.

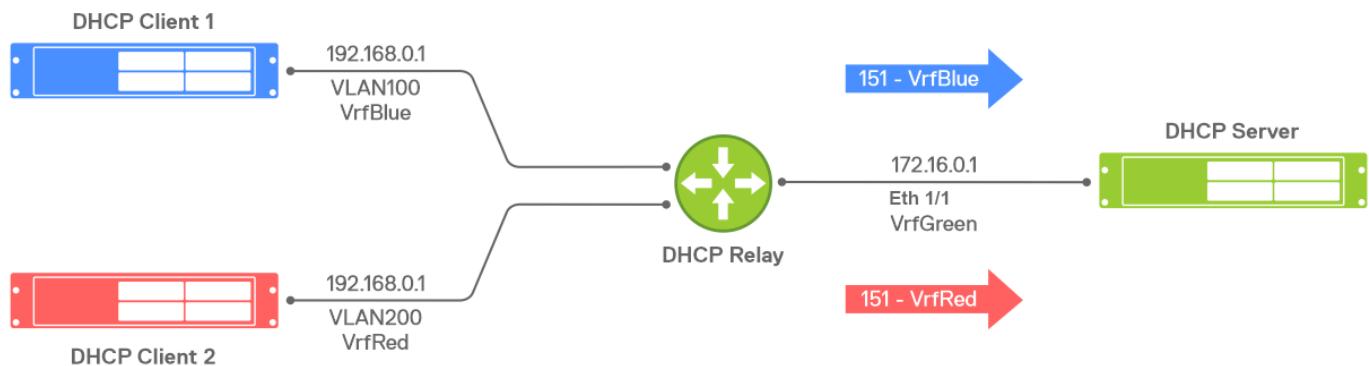
### Virtual subnet selection suboption

DHCP relay supports multiple clients on different VRFs which share overlapping IP addresses. In such VRF deployments, the DHCP server needs to be aware of the VRF of the client so that the address allocation can be done based on that VRF. To provide VRF information, DHCP relay includes the suboption 151 for DHCPv4 and suboption 68 for DHCPv6 as defined in RFC 6607.

The virtual subnet selection suboption (type 0) carries the ASCII VRFNAME configured on the incoming interface to which the client is connected. If the incoming interface is in default VRF, the sub option is not added to the relayed packet.

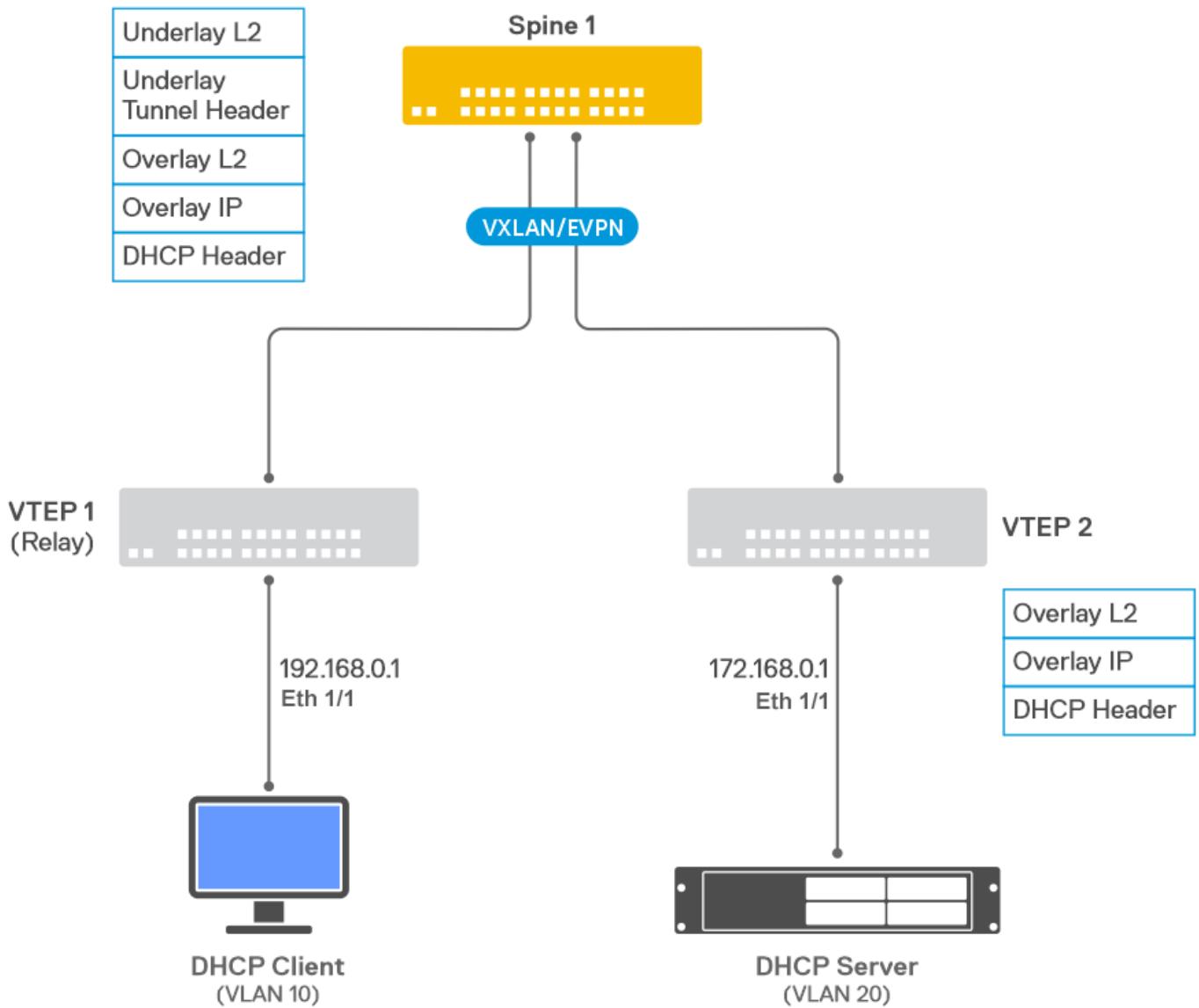
To ensure interoperability, enable the virtual subnet selection suboption only when the DHCP server supports address allocation based on VRF. Some servers may not recognize the suboption and may still allocate lease in the default VRF space. DHCP relay does not discard these replies from server.

In the following figure, both DHCP Client1 and DHCP Client2 use the same address space. If you configure virtual subnet selection suboption on the DHCP Relay switch, the relay device includes the suboption when sending the packet to the DHCP server.



### DHCP relay in a VXLAN deployment

You can configure DHCP relay in a VXLAN BGP EVPN deployment to provide DHCP services to EVPN clients or VMs. The following figure shows a typical deployment in VXLAN networks. The client and server can be in the same or different VRF domains.



The following illustrates the functioning of DHCP relay in a VXLAN deployment:

1. DHCP client is attached to VTEP1 on VLAN 10, which is bound to VrfRed.
2. DHCP relay is enabled on VTEP1 for VLAN 10.
3. DHCP server is on VTEP2 and is connected to VLAN 20, which is bound to VrfRed.
4. VTEP1 has BGP EVPN type-5 route to the DHCP server 172.16.0.1 that points to VXLAN tunnel next hop.
5. The DHCP relay forwards the incoming packet to 172.16.0.1 with giaddr set to 192.168.0.1. The relay is unaware of the VXLAN tunnels.
6. VTEP1 adds underlay Layer 2 tunnel headers and forwards the packet to the destination VTEP2.
7. VTEP2 removes the underlay Layer 2 and tunnel headers, and forwards the DHCP packet to the server.
8. DHCP server responds to giaddr 192.168.0.1. VTEP2 has BGP EVPN type-5 route to the relay agent IP address 192.168.0.1 that points to the tunnel next hop.
9. VTEP2 adds underlay Layer 2 tunnel headers and sends the response packet to VTEP1.
10. The DHCP relay agent on VTEP1 receives the response from server, removes option 82, and forwards the packet to the client on VLAN 10.

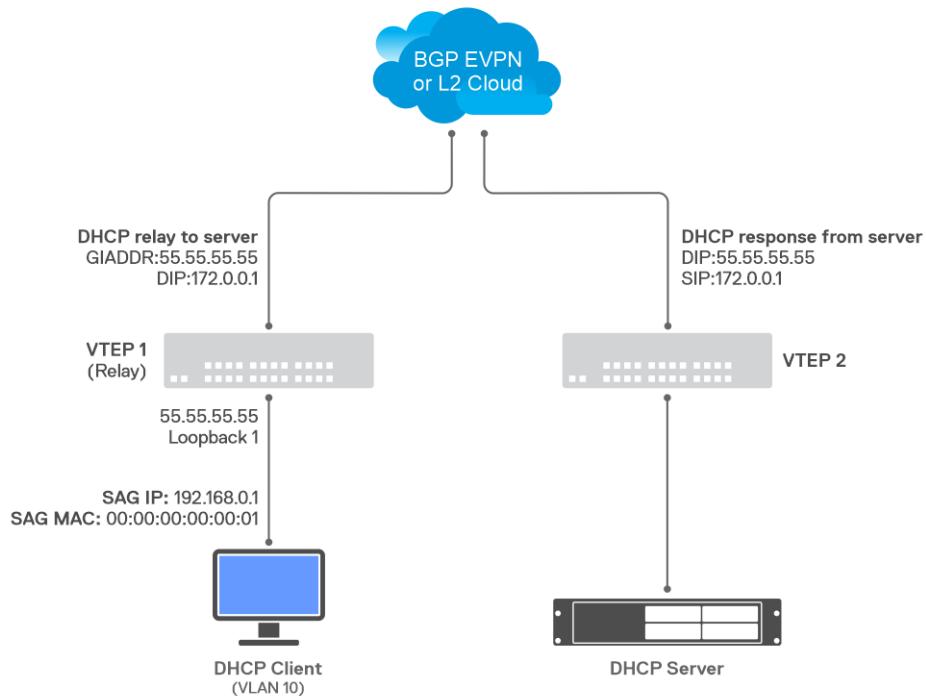
**(i) NOTE:** DHCP relay in BGP EVPN deployments is applicable to Layer 3 VNI configurations. For Layer 2 VNI configurations, there is no need for DHCP relay, as the client and server are directly reachable over extended VLAN.

#### DHCP relay and Static Anycast Gateway

Static Anycast Gateway (SAG) allows multiple switches to simultaneously route packets using a common gateway address in an active/active router configuration. Each switch is configured with the same set of virtual IP address and a virtual MAC address.

DHCP relay requires an IP address to identify the subnet of the downstream or client-facing interface. If the client interface is enabled for SAG, DHCP relay uses the SAG IPv4 address as the giaddr. If the associated SAG interface does not have any IP address that is assigned, relay agent discards the packet. As identical SAG IP address is configured on Leaf switches, the response from the server may land on a different leaf switch, and may not reach the leaf switch that relayed the DHCP packet. To avoid this issue, use link-selection option with the source interface.

In the following figure, 192.168.0.1 is used as the SAG gateway for VLAN10 on the leaf switch. DHCP relay is enabled on VLAN10. To relay a DHCP packet to the DHCP server, the giaddr field is set to 55.55.55.55. The DHCP server uses link-selection suboption 5 to identify the client subnet to be leased. The response from DHCP server is sent to the Loopback IP which is unique to the originating leaf switch.



### DHCP relay on unnumbered IPv4 interfaces

You can configure DHCP relay on unnumbered point-to-point links. The IPv4 unnumbered configuration enables Layer 3 processing without assigning an explicit IPv4 address.

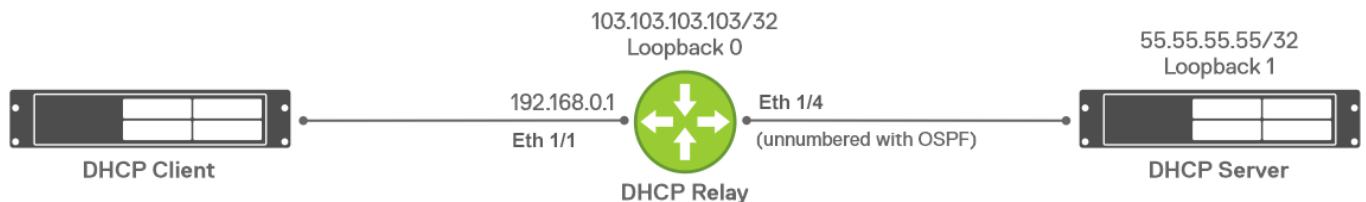
The unnumbered interface uses the IPv4 address of another interface that is already configured on the router. You can use the IPv4 unnumbered configuration to save network address space and simplify the switch configuration.

DHCP relay supports forwarding packets to a server through an IPv4 unnumbered interface with the following limitations:

- Only loopback interfaces are supported as donor interfaces.
- IPv4 unnumbered configuration is supported only on Ethernet and port channel interfaces.
- IPv4 unnumbered configuration is supported only on the default VRF.

**i** **NOTE:** Both ends of the link between the relay and the server must be configured as unnumbered interfaces. The client facing downstream interface must have an IPv4 address associate as the server must know the client subnet to assign the DHCP lease.

In the following figure, IPv4 unnumbered is configured on Eth1/4 interface, which is a point-to-point link between the relay and server. OSPFv2 is enabled on relay switch and server switch, and the loopback network addresses are advertised.



The client subnet is also advertised through OSPFv2 so that the server can reply to the relay switch. The giaddr in the relayed packet is set to 192.168.0.1. The source IPv4 address in the relayed packet is determined by the routing stack.

You can also enable link-selection in an IPv4 unnumbered setup. If the client subnet is not reachable from server, the giaddr is set to the Loopback1 address which is 103.103.103.103.

### Handling DHCPv4 packets with relay agent options

To support different network configurations, such as, cascading relays, the relay agent provides three different options to handle incoming DHCPv4 packets that already have relay agent options.

- **discard** - The relay agent discards the incoming packet (default).
- **append** - The relay agent appends its own set of relay options to the packet, leaving the incoming options intact. If the length of relay agent information exceeds the max limit of 255 bytes, the packet is discarded.
- **replace** - The relay agent removes the incoming options and adds its own set of options to the packet.

### Server identifier override sub-option

DHCPv4 relay supports server identifier override suboption 11 as defined in RFC5107. This suboption enables the relay to act as the DHCPv4 server so that unicast DHCPv4 packets come to the relay agent instead of directly going to the server. The relay can add appropriate suboptions on the unicast packets. The server identifier override suboption is automatically added when link-select suboption or VSS suboption is enabled.

If the DHCPv4 server does not support the server identifier suboption, then the unicast DHCPv4 packets from client are sent directly to server bypassing the relay agent.

This suboption is only applicable to DHCPv4 relay agent.

### Scalability

- You can enable a maximum of four relay addresses per interface.
- DHCPv4 and DHCPv6 relay can handle up to 2000 DHCP clients.
- You can enable a maximum of 4000 Layer 3 interfaces for DHCPv4 relay.
- You can enable a maximum of 4000 Layer 3 interfaces for DHCPv6 relay.

## Configure DHCP relay

To configure DHCP relay, follow these steps:

- Enable DHCP relay on the interface that you plan to use for relaying.

```
sonic(conf-if) # ip dhcp-relay dhcp-server-ip-address vrf vrf-name
```

- *dhcp-server-ip-address* - Enter the server address. You can add up to four addresses.
- *vrf vrf-name* - (Optional) Enter the VRF name.

- Enter source interface selection on an interface.

```
sonic(conf-if) # ip dhcp-relay source-interface interface
```

- (Optional) Enable link-selection suboption on an interface.

```
sonic(conf-if) # ip dhcp-relay link-select
```

- (Optional) To add VRF information in the DHCP packet that is sent to the DHCP server, specify the VRF-select option.

```
sonic(conf-if) # ip dhcp-relay vrf-select
```

- (Optional) Set the maximum hop limit.

```
sonic(conf-if-Vlan100) # ip dhcp-relay max-hop-count hop-count
```

- *hop-count* - Specify the hop count. The range is from 0 to 16. The default value is 10.

- (Optional) Specify how to handle a DHCP relay packet that comes from another relay agent.

```
sonic(conf-if) # ip dhcp-relay policy-action [discard | append | replace]
```

## Configure DHCPv6 relay

To configure DHCP relay, follow these steps:

- Enable DHCP relay on the interface that you plan to use for relaying.

```
sonic(conf-if)# ipv6 dhcp-relay dhcp-server-ipv6-address vrf vrf-name
```

- *dhcp-server-ipv6-address* - Enter the server IPv6 address. You can add up to four addresses.
- *vrf vrf-name* - (Optional) Enter the VRF name.

- Enter source interface selection on an interface.

```
sonic(conf-if)# ipv6 dhcp-relay source-interface interface
```

- (Optional) Set the maximum hop limit.

```
sonic(conf-if-Vlan100)# ipv6 dhcp-relay max-hop-count hop-count
```

- *hop-count* - Specify the hop count. The range is from 1 to 16. The default value is 10.

- (Optional) Specify how to handle a DHCP relay packet that comes from another relay agent.

```
sonic(conf-if)# ip dhcp-relay policy-action [discard | append | replace]
```

### View DHCP server address configuration

```
sonic# show ip dhcp-relay brief

Interface DHCP Helper Address

Eth1/1 30.1.1.1
Eth1/1 40.1.1.1
```

```
sonic# show ipv6 dhcp-relay brief

Interface DHCP Helper Address

Eth1/1 300::1
Eth1/1 400::1
Eth1/1 500::1
```

### View the DHCP relay statistics on an interface

```
sonic# show ip dhcp-relay statistics
String Interface name

sonic# show ip dhcp-relay statistics Vlan100
BOOTREQUEST messages received by the relay agent : 4
BOOTREQUEST messages forwarded by the relay agent : 2
BOOTREPLY messages forwarded by the relay agent : 0
DHCP DISCOVER messages received by the relay agent : 1
DHCP OFFER messages sent by the relay agent : 0
DHCP REQUEST messages received by the relay agent : 1
DHCP ACK messages sent by the relay agent : 0
DHCP RELEASE messages received by the relay agent : 0
DHCP DECLINE messages received by the relay agent : 0
DHCP INFORM messages received by the relay agent : 0
DHCP NACK messages sent by the relay agent : 0
Total number of DHCP packets dropped by the relay agent : 2
Number of DHCP packets dropped due to an invalid opcode : 0
Number of DHCP packets dropped due to an invalid option : 0
```

```
sonic# show ipv6 dhcp-relay statistics
String Interface name

sonic# show ipv6 dhcp-relay statistics Vlan100
DHCPv6 SOLICIT messages received by the relay agent : 1
DHCPv6 ADVERTISEMENT messages sent by the relay agent : 1
```

```

DHCPv6 REQUEST messages received by the relay agent : 1
DHCPv6 REPLY messages sent by the relay agent : 1
DHCPv6 CONFIRM messages received by the relay agent : 0
DHCPv6 RELEASE messages received by the relay agent : 0
DHCPv6 DECLINE messages received by the relay agent : 0
DHCPv6 REBIND messages received by the relay agent : 0
DHCPv6 RECONFIGURE messages sent by the relay agent : 0
DHCPv6 INFO-REQUEST messages received by the relay agent : 0
DHCPv6 RELAY-REPLY messages received by the relay agent : 2
DHCPv6 RELAY-FORWARD messages sent by the relay agent : 2
Total number of DHCPv6 packets dropped by the relay agent : 0
Number of DHCPv6 packets dropped due to an invalid opcode : 0
Number of DHCPv6 packets dropped due to an invalid option : 0

```

#### View the relay configuration on a given interface

```

sonic# show ip dhcp-relay detailed Vlan100

Relay Interface: Vlan100
Server Address: 112.0.0.2
Server VRF: VrfRed
Source Interface: Loopback1
Link Select: enable
VRF Select: enable
Max Hop Count: 10
Policy Action: Discard

```

```

sonic# show ipv6 dhcp-relay detailed Vlan100

Relay Interface: Vlan100
Server Address: 2000::2
Server VRF: VrfRed
Source Interface: Not Configured
VRF Select: enable
Max Hop Count: 10

```

## DHCP relay circuit-id option

When a DHCP relay agent forwards DHCP requests from a client to a DHCP server, it can include encoded circuit ID information based on the interface on which the DHCP client packet is received. The circuit ID is used to relay DHCP server response packets back to the originating client interface. In addition, the DHCP server may also use the circuit ID to assign an IP address and other parameters.

**(i) NOTE:** Option 82 for sending DHCP relay agent information is applied only on DHCPv4 packets.

To configure the circuit-id format for DHCPv4 relay option 82, use the `ip dhcp-relay circuit-id` command. The command allows you to specify the format in which circuit ID information is sent:

- `%p` — Sends the name of the interface name on which request was received; for example, `Vlan100` (default).
- `%h:%p` — Sends the hostname of the switch followed by interface name; for example, `sonic-acc-sw-01:Vlan100`.
- `%i` — Sends the name of the physical interface on which request was received; for example, `Eth1/2`. Use this option to identify the physical interface when multiple clients are connected in a VLAN. The DHCP server can then assign DHCP leases according to individual client interfaces. If a VLAN name is sent in the circuit ID information, all clients will have the same circuit ID.

The default circuit ID format is `%p`. Configure a circuit-ID format for an interface on a DHCP relay switch that receives DHCP client packets. To delete the configured circuit-ID format and return to the default, enter the `no ip dhcp-relay circuit-id` command.

```
sonic(conf-if)# ip dhcp-relay circuit-id [%p | %h:%p | %i]
```

## DHCP Option 82 — circuit ID examples

```
Agent-Information Option 82, length 28:
 Circuit-ID SubOption 1, length 7: "Vlan100"
 Remote-ID SubOption 2, length 17: 52:54:00:c1:65:6b
```

```
Agent-Information Option 82, length 43:
 Circuit-ID SubOption 1, length 22: "sonic-acc-sw-01:Vlan100"
 Remote-ID SubOption 2, length 17: 52:54:00:c1:65:6b
```

## DHCP snooping

**i** **NOTE:** DHCP snooping is available only in the Enterprise Standard, Enterprise Premium, and Edge Standard bundles. DHCP snooping is not available in the Cloud Standard and Cloud Premium bundles.

DHCP snooping is a security feature that enables switches in a network to monitor DHCP control messages. Using the control messages, a switch can identify rogue DHCP servers and clients in a network.

When you enable DHCP snooping, the switch starts monitoring DHCP control packets both from DHCP servers and clients. The system uses this information to build a database.

With DHCP snooping, there are two types of interfaces. They are trusted and untrusted interfaces. You connect DHCP clients to untrusted interfaces and DHCP servers to trusted interfaces.

When the switch receives DHCP messages from clients on untrusted ports that it forwards the packets to the trusted ports in the same VLAN. When you configure ports that connect to DHCP servers as trusted, the system drops any DHCP-server-to-client messages that it receives on untrusted interfaces.

By checking the source MAC address in the Ethernet header with the client MAC address in the DHCP header, DHCP snooping minimizes malicious DHCP clients from acquiring a DHCP lease.

### Configuration notes

- DHCP snooping is supported for IPv4 and IPv6.
- DHCPv6 snooping works only with DHCPv6 stateful server.
- Enable DHCP snooping globally and on specific VLANs.
- Configure ports within the VLAN to be trusted or untrusted.
- By default, all ports are untrusted.
- Connect DHCP servers through trusted ports.
- Connect DHCP clients through untrusted ports.
- On untrusted interfaces, the switch drops DHCP packets if the source MAC address does not match the client hardware address. You can disable this behavior disabling the Verify MAC address feature. Use this feature for DHCP relay and DHCP unicast request packets that are routed.
- When you enable DHCP snooping, the Verify MAC address feature is enabled by default.
- DHCP Snooping is not applied to VLANs on which snooping is not enabled.
- You can configure a manual entry to the DHCP snooping binding table.
- Before enabling DHCP snooping, remove the DHCP Layer 3 CoPP rule and install the DHCP Layer 2 CoPP rule. For more details, see the **DHCP Layer 2 CoPP configuration** section.

## DHCP Layer 2 CoPP configuration

For DHCP snooping to work, perform this configuration:

- Uninstall DHCP Layer 3 CoPP rules. This is a default rule.

```
sonic# configure terminal
sonic(config)# policy-map copp-system-policy type copp
sonic(config-policy-map)# no class copp-system-dhcp
```

- Install DHCP Layer 2 CoPP rule for DHCP snooping.

```
sonic# configure terminal
sonic(config)# policy-map copp-system-policy type copp
```

```
sonic(config-policy-map)# class copp-system-dhcp12
sonic(config-policy-map-flow)# set copp-action copp-system-dhcp
```

## Configure DHCP snooping for IPv4

To configure DHCP snooping, use the following procedure:

1. Enable DHCP snooping globally.

```
sonic(config)# ip dhcp snooping
```

2. Configure interfaces that are connected to the DHCP server as trusted.

```
sonic(config-if)# ip dhcp snooping trust
```

3. Enable DHCP snooping on a VLAN or a VLAN list.

```
sonic(config)# ip dhcp snooping vlan {vlan-id | vlan-list}
```

4. (Optional) Disable DHCP source MAC address verification.

```
sonic(config)# no ip dhcp snooping verify mac-address
```

5. (Optional) Create a static entry to the DHCP snooping binding table.

```
sonic(config)# ip source binding source-ip-address source-mac-address vlan vlan-id
interface interface-name
```

 **NOTE:** To remove a static entry, use the no form of the ip source binding command.

## Clear entries from the DHCP snooping binding table for IPv4

Use the following procedure to clear all or a specific dynamic entry:

- Clear all dynamic IP DHCP snooping binding entries:

```
sonic(config)# clear ip dhcp snooping binding
```

- Clear a specific dynamic IP DHCP snooping binding entry:

```
sonic(config)# clear ip dhcp snooping binding source-ip-address source-mac-address
vian vlan-id interface interface-name
```

- Clear DHCP snooping statistics:

```
sonic# clear ip dhcp snooping statistics
```

## View DHCP snooping information for IPv4

- View general information about DHCP snooping:

```
sonic# show ip dhcp snooping
```

- View the DHCP snooping binding database:

```
sonic# show ip dhcp snooping binding
```

- View DHCP snooping statistics:

```
sonic# show ip dhcp snooping statistics
```

## View DHCP snooping information-Examples for IPv4

View general information about DHCP snooping:

```
sonic# show ip dhcp snooping

DHCP snooping is Enabled
DHCP snooping source MAC verification is disabled
DHCP snooping is enabled on the following VLANs: 10,20

Interface Trusted

Ethernet1 Yes
...
(Config)#show ip dhcp snooping binding

Total number of bindings: 2
Total number of Tentative bindings: 0

MAC Address IP Address VLAN Interface Type Lease (Secs)
----- ----- ----- ----- -----
00:00:00:00:00:01 1.1.1.1 10 Ethernet0 STATIC -
00:00:A8:5F:34:52 192.168.10.39 20 Ethernet2 DYNAMIC 86396
```

View DHCP snooping statistics:

```
sonic# show ip dhcp snooping statistics

Interface MAC Verify Client Ifc DHCP Server
 Failures Mismatch Msgs Recvd

Ethernet1 0 0 0
Ethernet2 0 0 0
Ethernet3 0 0 0
Ethernet4 0 0 0
Ethernet5 0 0 0
Ethernet6 0 0 0
Ethernet7 0 0 0
Ethernet8 0 0 0
```

## Configure DHCP snooping for IPv6

To configure DHCP snooping, use the following procedure:

1. Enable DHCP snooping globally.

```
sonic(config)# ipv6 dhcp snooping
```

2. Configure interfaces that are connected to the DHCP server as trusted.

```
sonic(config-if)# ipv6 dhcp snooping trust
```

3. Enable DHCP snooping on a VLAN or a VLAN list.

```
sonic(config)# ipv6 dhcp snooping vlan {vlan-id | vlan-list}
```

4. (Optional) Disable DHCP source MAC address verification.

```
sonic(config)# no ipv6 dhcp snooping verify mac-address
```

5. (Optional) Create a static entry to the DHCP snooping binding table.

```
sonic(config)# ipv6 source binding source-ip-address source-mac-address vlan vlan-id
 interface interface-name
```

 **NOTE:** To remove a static entry, use the no form of the ip source binding command.

## Clear entries from the DHCP snooping binding table for IPv6

Use the following procedure to clear all or a specific dynamic entry:

- Clear all dynamic IP DHCP snooping binding entries:

```
sonic(config)# clear ipv6 dhcp snooping binding
```

- Clear a specific dynamic IP DHCP snooping binding entry:

```
sonic(config)# clear ipv6 dhcp snooping binding source-ip-address source-mac-address
vlan vlan-id interface interface-name
```

- Clear DHCP snooping statistics:

```
sonic# clear ipv6 dhcp snooping statistics
```

## View DHCP snooping information for IPv6

- View general information about DHCP snooping:

```
sonic# show ipv6 dhcp snooping
```

- View the DHCP snooping binding database:

```
sonic# show ipv6 dhcp snooping binding
```

- View DHCP snooping statistics:

```
sonic# show ipv6 dhcp snooping statistics
```

### View DHCP snooping information-Examples for IPv6

View general information about DHCP snooping:

```
sonic# show ipv6 dhcp snooping

DHCP snooping is Enabled
DHCP snooping source MAC verification is disabled
DHCP snooping is enabled on the following VLANs: 10,20

Interface Trusted
----- -----
Ethernet1 Yes
...
(Config)#show ip dhcp snooping binding

Total number of bindings: 2
Total number of Tentative bindings: 0

MAC Address IP Address VLAN Interface Type Lease (Secs)
----- ----- ----- ----- ----- -----
00:00:00:00:00:01 1::1 10 Ethernet0 STATIC -
00:00:A8:5F:34:52 2::1 20 Ethernet2 DYNAMIC 86396
```

## DHCP snooping and DHCP relay on same VLAN

In some deployments, the access switch requires both DHCP snooping and Layer 3 DHCP relay to be enabled on the same VLAN simultaneously. DHCP snooping maintains the client bindings and DHCP relay forwards the packet to the DHCP server.

You can enable both these features allowing PAC applications to get the client IP address for authentication. Enabling both features also enable Layer 3 connectivity to the DHCP server through relay.

When both DHCP snooping and DHCP relay are enabled on same VLAN, packets are processed in the following sequence :

Client packets:

- DHCP snooping intercepts the client packet, processes it, and passes the packet to DHCP relay.
- DHCP snooping does not forward the client packet on trusted ports, if any.
- DHCP relay handles the client packet as per the relay configuration and forwards the packet to DHCP servers.

Server packets:

- DHCP relay intercepts the server packet, processes it, and passes the packet to DHCP snooping.
- DHCP relay does not forward the server response to the client.
- DHCP snooping validates the server response and forwards the DHCP packet to the client.
- DHCP snooping skips trusted port validation.

## Third party containers

Enterprise SONiC supports the installation, management, and upgrade of third-party containers.

In Enterprise SONiC, the third-party container (TPC) feature allows you to install third-party components as containers that are loaded and integrated as part of the SONiC system without having to rebuild the Enterprise SONiC image. The TPC feature provides a mechanism to install and manage custom Docker containers in the SONiC operating system.

The TPC feature supports the following operations in a third-party container life cycle:

- Download and install a Docker image.
- Pull the Docker image from the docker registry.
- Create and run a Docker container.
- Store docker-specific data in its own file system or the host-mounted file system.
- Save a Docker snapshot as an image to be exported elsewhere.
- Stop or uninstall a Docker container.
- Remove a Docker image.
- Upgrade an existing Docker container.

### Usage notes

- A TPC image can be upgraded independently of an Enterprise SONiC image upgrade. Seamless TPC upgrade with container data backup and restore is supported.
- An Enterprise SONiC image upgrade seamlessly migrates all currently installed third-party containers into the newly installed image.
- All third-party containers automatically start during fast, warm, and cold system reboots.
- You can install and upgrade third-party containers from several sources: HTTP servers, SCP/SFTP servers, locally installed media, and an external Docker registry.
- You can run the TPC framework on a statically configured or dynamically configured VRF.
- The TPC feature is supported on switches with a minimum 32GB disk. The overall TPC disk usage is not allowed to be more than 20% of the total disk space.
- The overall TPC memory usage is not allowed to be more than 20% of the total system memory. A single TPC is not allowed to consume more than 20% of the overall TPC memory.

**i** **NOTE:** Enterprise SONiC is a containerized micro-services architecture. Standard SONiC containers have dependencies which can be interrupted if certain micro-services are not up. The only containers that you can stop and start without affecting Enterprise SONiC operation are third party containers.

## Install a TPC image

### Installation notes

- When you install a TPC container into Enterprise SONiC, you can create the container either from an image file or pull it from an external Docker registry.
- You can install one or more TPC images on the system. When Enterprise SONiC is up, bring up the TPC instance by running the `tpcm install` command.
- TPC image installation creates the SONiC service that is required to bring up the TPC container. It is necessary to load a TPC container image only once. On later reboots, the TPC service starts automatically by the systemd manager in the system.

- The TPC service file is generated for each TPC container. The generated service is stored in the TCPM storage folder and also installed in the systemd service folder.
- The TPC container installation process performs these steps:
  1. Download the TPC image from a source.
  2. Install the TPC image in Enterprise SONiC.
  3. Create a systemd service and configure the resource for the newly installed TPC image.
  4. Create a container for the newly installed TPC image.
  5. Bring up the container instance by starting the TPC service.

### TPC image installation

Install a TPC image from one of the following sources:

- HTTP server — Install from an HTTP server.
- SCP path — Copy from a remote server using the SCP protocol.
- SFTP server — Copy from a remote server using the SFTP protocol.
- Media path — Copy from a local media folder.
- Docker hub — Download from a remote Docker registry.
- Local image — Use an existing docker image.
- Install the TPC image file from an external HTTP/HTTPS server.

```
sonic# tpcm install name tpc-container-name url url [vrf-name vrf-name] [args docker-arguments] [cargs container-arguments] [start-after-system-ready {True | False}]
```

- *name tpc-container-name* — Enter the name of the container (255 characters maximum).
- *url url* — Enter the URL of the server where the TPC image is stored.
- *vrf-name vrf-name* — (Optional) Enter the name of the VRF in which the container runs (15 characters maximum in the format: *Vrfname*); by default, a TPC uses the default VRF.
- *args docker-arguments* — (Optional) Enter standard docker arguments.
- *cargs container-arguments* — (Optional) Enter container arguments to specify additional arguments for a container's init process or a script; for example, “*-path.rootfs=/host*”.
- *start-after-system-ready {True | False}* — (Optional) Specify whether to bring up the container after system startup (Default: True).

For example:

```
sonic# tpcm install name mydocker url http://myserver/path/mydocker.tar.gz
```

```
sonic# tpcm install name mydocker url http://myserver/path/mydocker.tar.gz args " -e TESTENV=TESTVALUE"
```

- Load the TPC image file from an external server using SCP or SFTP.

```
sonic# tpcm install name tpc-container-name {scp | sftp} server-name username username password password filename tpc-image-path [vrf-name vrf-name] [args docker-arguments] [cargs container-arguments] [start-after-system-ready {True | False}]
```

- *name tpc-container-name* — Enter the name of the container (255 characters maximum).
- *{scp | sftp} server-name username username password password filename tpc-image-path* — Enter the SCP or SFTP server name, login credentials, and TCP image file path.

For example:

```
sonic# tpcm install name mydocker scp myserver username myuser password paswd filename /path/mydocker.tar.gz
```

- Load the TPC image file from a local media path.

```
sonic# tpcm install name tpc-container-name file tpc-image-path [vrf-name vrf-name] [args docker-arguments] [cargs container-arguments] [start-after-system-ready {True | False}]
```

- *name tpc-container-name* — Enter the name of the container (255 characters maximum).
- *file tpc-image-path* — Enter the path on the local installed media where the TPC image is stored.

For example:

```
sonic# tpcm install name mydocker file /media/usb/path/mydocker.tar.gz
```

```
sonic# tpcm install name mydocker url http://myserver/path/mydocker.tar.gz args " -e TESTENV=TESTVALUE"
```

- Load the TPC image file from an external Docker repository.

```
sonic# tpcm install name tpc-container-name pull image-name [:tag-name] [vrf-name vrf-name] [args docker-arguments] [cargs container-arguments] [start-after-system-ready {True | False}]
```

- *name tpc-container-name* — Enter the name of the container (255 characters maximum).
- *pull image-name* [:*tag-name*] — Enter the name of a TPC image in a Docker repository.

For example:

```
sonic# tpcm install name mydocker pull ubuntu:latest
```

- Load the TPC image file from an existing Docker image.

```
sonic# tpcm install name tpc-container-name image image-name [:tag-name] [vrf-name vrf-name] [args docker-arguments] [cargs container-arguments] [start-after-system-ready {True | False}]
```

- *name tpc-container-name* — Enter the name of the container (255 characters maximum).
- *image image-name* [:*tag-name*] — Enter the name of an existing Docker image.

For example:

```
sonic# tpcm install name mydocker image ubuntu:latest
```

## Using Management VRF

By default, third-party containers use the default VRF. However, if a Management VRF (see [Management VRF](#)) has been created in the system, installed dockers and third-party containers do not have connectivity to an external network. To connect to an external network, a third-party container must use the Management VRF namespace. To use the Management VRF, specify the Management VRF during TPC image installation.

```
sonic# tpcm install name tpc-container-name file tpc-image-path vrf-name mgmt
```

For example:

```
sonic# tpcm install name mydocker file /path/docker-tpcimage.gz vrf-name mgmt
```

## Uninstall a third-party container

The TPC uninstallation process performs these steps:

1. Stop running a TPC container by stopping the corresponding TPC service.
2. Remove the TPC service file from the SONiC systemd service manager.
3. Remove the third-party container from the SONiC system.
4. Remove the TPC image from the SONiC system.

After a third-party container and its associated service are removed, the removed TPC services do not automatically start at the next system reboot.

To remove a third-party container, enter the `tpcm uninstall` command.

```
sonic# tpcm uninstall name tpc-container-name [clean-data {Yes | No}]
```

- *name tpc-container-name* — Enter the name of the container.
- *clean-data {Yes | No}* — (Optional) Specify whether to remove container data (Yes) or to leave it in the system (No). Default: No.

For example:

```
sonic# tpcm uninstall name mydocker clean-data Yes
```

## Upgrade TPC image

You can upgrade a TPC image and update the configuration of an installed image, including:

- Docker arguments, such as `--privileged`, `--network`, and `--memory` which is used to reconfigure the memory limit
- Container arguments, such as `"-path.rootfs=/host"`

To upgrade a third-party container, enter the `tpcm upgrade` command.

```
sonic# tpcm upgrade name tpc-container-name [url url] [{scp | sftp} server-name username username password filename tpc-image-path] [args docker-arguments] [cargs container-arguments] [skip-data-migration {Yes | No}]
```

- `name tpc-container-name` — Enter the name of the container (255 characters maximum).
- `url url` — (Optional) Enter the URL of the server where the TPC image is stored.
- `{scp | sftp} server-name username username password filename tpc-image-path` — (Optional) Enter the SCP or SFTP server name, login credentials, and TCP image file path.
- `args docker-arguments` — (Optional) Enter standard docker arguments.
- `cargs container-arguments` — (Optional) Enter container arguments to specify additional arguments for a container's init process or a script; for example, `"-path.rootfs=/host"`.
- `skip-data-migration {Yes | No}` — (Optional) Specify whether to migrate and retain container data (No) or to not migrate the data (Yes). Default: No.

For example:

```
sonic# tpcm upgrade name mydocker image ubuntu:latest
```

```
sonic# tpcm upgrade name mydocker sftp myserver username myuser password passwd filename mydocker.tar.gz skip_data_migration yes
args " -e TESTENV=TESTVALUE"
```

## Update TPC image

You can update the configuration of an installed image, including:

- VRF in which the TPC runs and restarts the TPC.
- Start-after-system-ready setting — Brings up the container after system startup.
- Memory capacity of the TPC — Reconfigures the memory limit and does not restart the TPC.

To update a TPC configuration, enter the `tpcm update` command.

```
sonic# tpcm update name tpc-container-name [memory memory-value] [vrf-name vrf-name]
[start-after-system-ready {True | False}]
```

- `name tpc-container-name` — Enter the name of the container (255 characters maximum).
- `memory memory-value` — Enter the memory that is allocated to a TPC with one of the b k m g characters or a KB, MB, or GB value; or a K, M, or G value; or enter a memory unit without any postfix character to be considered as a simple byte value. Default: The overall memory limit for a TPC is 20% of the total system memory.
- `vrf-name vrf-name` — (Optional) Enter the name of the VRF in which the container runs (15 characters maximum in the format: `Vrfname`); by default, a TPC uses the default VRF.
- `start-after-system-ready {True | False}` — (Optional) Specify whether to bring up the container after system startup. Default: True.

For example:

```
sonic# tpcm update name TEST memory 200M vrf-name "mgmt" start-after-system-ready False
sonic# tpcm update name TPC2 memory 1G vrf-name "mgmt" start-aftersystem-ready False
```

```
sonic# tpcm update name TPC3 memory 500M vrf-name "mgmt" start-aftersystem-ready False
```

## Update overall TPC disk limit

Use the `tpcm update disk-limit` command to update the overall disk limit for all installed third-party containers. Enter a value to be used to set the maximum disk space allowed for all TPCs on the switch. The disk-value must be a unit with one of the postfix characters: G M K g m k. The disk-value cannot be a single decimal number.

```
sonic# tpcm update disk-limit disk-value
```

Examples:

```
sonic# tpcm update disk-limit 8G
```

```
sonic# tpcm update disk-limit 4000M
```

## View TPC images

To display the third-party containers installed on the switch, use the `show tpcm list` and `show tpcm name` commands.

```
sonic# show tpcm list
```

```
sonic# show tpcm list
CONTAINER NAME IMAGE TAG VRF CONFIGURED/RUNNING STATUS
TEST mydocker:latest default/default Up 8 seconds
```

```
sonic# show tpcm name telegraf
TPC docker args:
--network=host -v /etc/sonic/frr:/etc/sonic/frr -v /etc/resolv.conf:/etc/resolv.conf --
hostname=sonic
TPC container CMD:
```

```
TPC configs :
--cpu-period=100000 --cpu-quota=20000 --cpu-shares=0 --cpus=0 --memory=618m --memory-
swap=618m
--memory-reservation=0
TPC service file:
```

```
[Unit]
Description=telegraf docker
After=docker.service
```

```
[Service]
ExecStartPre=/usr/local/bin/tpc.sh start telegraf telegraf:37141b42
ExecStart=/usr/local/bin/tpc.sh wait telegraf telegraf:37141b42
ExecStop=/usr/local/bin/tpc.sh stop telegraf telegraf:37141b42
StandardOutput=syslog
StandardError=syslog
Restart=on-failure
RestartSec=60
```

```
[Install]
WantedBy=tpcm.service
WantedBy=tpcm-user.target
```

```
TPC service drop in file:
[Unit]
After=systemready.service
```

```
TPC VRF name :
default
```

## L2 and L3 switch profiles

Enterprise SONiC supports L2 and L3 switch profiles to optimize the switch for L2 and L3 deployments. By default, the L3 switch profile is installed.

- Use the L2 profile for scalable L2 deployments to configure a larger MAC table size.
- Use the L3 profile for most switch deployments to configure a larger L3 table size.

**(i) NOTE:** Dell Technologies does not recommend that you use the L2 switch profile because it deactivates some L3 features, such as VXLAN. If a switch requires a larger MAC address table, use the `route scale` command. For more information, see [L2/L3 host and route scaling](#).

The differences between the L2 and L3 profiles are summarized in the following table:

**Table 12. Differences between L2 and L3 switch profiles**

| L2 profile (Layer 2 switch)                          | L3 profile (Layer 3 router)                        |
|------------------------------------------------------|----------------------------------------------------|
| Default Spanning-tree protocol is Rapid PVST+        | Spanning tree is not enabled by default            |
| VLAN 1 is present by default                         | There is no VLAN in the default configuration      |
| All interfaces are access ports in VLAN 1 by default | All ports are non-switchport interfaces by default |

## View the active switch profile

To view the active profile and available profile names, use the `show switch-profiles` command.

```
sonic# show switch-profiles
Factory Default: 13

Profile Name Description

12 Layer 2 Switch Configuration
13 Layer 3 Router Configuration
sonic#
```

## Change the switch profile

To modify the factory default profile, use the `factory default profile` command in CONFIGURATION mode. When changing profiles, configurations are deleted and the switch reboots, which causes all Enterprise SONiC application services to restart.

```
sonic# configure terminal
sonic(config)# factory default profile 12
Device configuration will be erased. You may lose connectivity.
Continue? [y/N]: y
Applying factory default configuration.
This may take 120--180 seconds and also result in a reboot.
sonic(config) #
```

# L2/L3 host and route scaling

To reconfigure the maximum number of L2 and L3 hosts and L3 route prefixes that are supported in the unified forwarding table (UFT) on the switch, use the `switch-resource` command. L2/L3 host and L3 route scaling allows you to reallocate switch resources in either of two modes:

**Route max mode** Increases the maximum number of IPv4 and IPv6 route prefixes supported in the forwarding table.

**Hosts layer2-layer3 mode** Increases the maximum number of L2 and L3 hosts that are supported in the forwarding table.

```
sonic(config)# switch-resource
sonic(config-switch-resource)# route-scale {routes {max | max-v6} | hosts {layer2-layer3 | layer2-layer3-balanced}}
```

 **NOTE:** The `route-scale max-v6` profile is not supported on Dell PowerSwitch platforms.

## Usage notes

- For the switch-specific maximum number of L2 hosts and L3 routes that are supported by default and in route-scale modes, see the *Enterprise SONiC Compatibility Matrix*.
- The Route max, Host layer2-layer3, and Host layer2-layer3-balanced modes apply only to the default L3 port profile. These route-scale modes do not apply to the default L2 port profile.
- The Route max mode is supported only on S5232F-ON, S5248F-ON, and S5296F-ON switches.
- The Hosts layer2-layer3 mode is supported only on S5232F-ON, S5248F-ON, and S5296F-ON switches.
- The Hosts layer2-layer3-balanced mode is supported only on Z9432F-ON and S5448F-ON switches. The hosts layer2-layer3-balanced mode is supported only in the Enterprise premium bundle and is not required in the Enterprise standard bundle.
- For the reallocation of switch resources to take effect, you must save the configured route-scale setting and reboot the switch.
- You can configure route scaling for only one mode at a time — either Route max mode or Hosts layer2-layer3/Hosts layer2-layer3-balanced mode. To configure a different route-scale mode, you must first unconfigure the currently enabled mode.
- When you enable a route-scale mode on an S5232F-ON, S5248F-ON, or S5296F-ON switch, the maximum number of supported IPv4 and IPv6 route prefixes, and L2/L3 hosts, increases from the defaults.
- When you scale route prefixes:
  - In the default and `routes max` mode, both IPv4 and IPv6 routes can co-exist in the forwarding table and scale to their maximum numbers. However, in the `hosts layer2-layer3` modes, the number of scalable routes is a one-dimensional number; that is, both IPv4 and IPv6 routes do not scale to the numbers in the L3 route prefix and L2/L3 host scaling tables simultaneously.
  - In host layer2-layer3 mode, a combination of IPv4 and IPv6 routes is supported. If you use only IPv4 routes, you can scale up to 160K routes. If you use only IPv6 routes, you can scale up to 65K routes. If you use both IPv4 and IPv6 routes, the maximum number of IPv4 or IPv6 routes depends on the number of routes configured for the other L3 protocol. For example, if 32K IPv6 routes are in use, the maximum number of IPv4 routes scales down to 80K.
  - Because L2/L3 hosts are entered in hardware as hash tables, the maximum number of supported hosts may vary due to hash collision.
  - The maximum number of scaled routes is applied to both underlay and overlay routes.

**Table 13. Example: L3 route prefix scaling on S5232F-ON, S5248F-ON, and S5296F-ON**

| Prefix type                    | Default | Route max mode | Hosts layer2-layer3 mode |
|--------------------------------|---------|----------------|--------------------------|
| IPv4 only                      | 81k     | 160k           | 160k                     |
| IPv6 only less than 64 bits    | 32k     | 65k            | 65k                      |
| IPv6 only greater than 64 bits | 25k     | 65k            | 65k                      |

**Table 14. Example: L2/L3 host scaling on S5232F-ON, S5248F-ON, and S5296F-ON**

| Host type | Default | Route max mode | Hosts layer2-layer3 mode |
|-----------|---------|----------------|--------------------------|
| L2 MAC    | 40k     | 32k            | 160k                     |

**Table 14. Example: L2/L3 host scaling on S5232F-ON, S5248F-ON, and S5296F-ON (continued)**

| Host type | Default           | Route max mode   | Hosts layer2-layer3 mode |
|-----------|-------------------|------------------|--------------------------|
| L3 ARP/ND | 32k ARP or 16k ND | 16k ARP or 8k ND | 16k ARP or 8k ND         |

**Table 15. Example: L2/L3-balanced host scaling on Z9432F-ON and S5448F-ON**

| Host type | Default | Hosts layer2-layer3-balanced mode |
|-----------|---------|-----------------------------------|
| L2 MAC    | 32k     | 64k                               |
| L3 ARP    | 30k     | 30k                               |
| L3 ND     | 16k     | 16k                               |

**Configure L3 route prefix scaling**

```
sonic(config)# switch-resource
sonic(config-switch-resource)# route-scale routes max
sonic(config-switch-resource)# exit
sonic(config)# exit
sonic# write memory
sonic# reboot
```

**Configure L2/L3 host scaling**

```
sonic(config)# switch-resource
sonic(config-switch-resource)# route-scale hosts layer2-layer3
sonic(config-switch-resource)# exit
sonic(config)# exit
sonic# write memory
sonic# reload
```

**Unconfigure L3 route prefix or L2/L3 host scaling and return to the defaults**

```
sonic(config)# switch-resource
sonic(config-switch-resource)# no route-scale {routes {max | max-v6} | hosts layer2-
layer3}
sonic(config-switch-resource)# exit
sonic(config)# exit
sonic# write memory
sonic# reload
```

**View route prefix and L2/L3 host scaling**

```
sonic# show switch-resource route-scale
Configured hosts : layer2-layer3
```

**Reserve host entries of local neighbors**

In an EVPN-based leaf-spine topology, when hosts connected to all leaf nodes come up simultaneously, there is a possibility that ARP entries from both local and remote nodes are programmed in the same host table in hardware. Although the ARP addresses learned by ARP flooding are temporary, if you have scaled the number of host addresses, the table can still fill up with transient remote ARP entries.

If the host table becomes full, locally learned ARP addresses may not have space and cannot be programmed in hardware. To avoid having a situation in which a full host table due to transient remote ARP entries rejects local-learned addresses, preconfigure the host table capacity for ARP entries from locally connected neighbors using the `ip reserve local neigh` command. Enter the maximum number of local neighbor IP addresses to reserve, from 0 to 32000; default 0.

```
sonic(config)# ip reserve local-neigh number
```

 **NOTE:** Reserving host table space for ARP entries from locally connected neighbors is a software-based limit check. There is no hardware partitioning.

To unconfigure the number of reserved local neighbor IP addresses, enter the `no ip reserve local-neigh` command.

To view the current number of reserved local neighbor IP addresses, enter the `show running-configuration` command.

```
sonic# show running-configuration | grep local-neigh
ip reserve local-neigh 1000
```

## Cut-through switching

Use cut-through switching to achieve lower latency in packet switching for time-sensitive cases. Cut-through switching is a global or switch-level configuration that is applied on all ports on a switch. By default, a switch is configured to use store-and-forward packet switching. In traditional store-and-forward (SF) mode, the larger the packet size, the higher the latency.

**(i) NOTE:** Cut-through switching is supported only on S5232F-ON, S5248F-ON, S5296F-ON, Z9332F-ON, Z9432F-ON, and Z9664F-ON switches.

### Benefits of cut-through switching

- In storage networks, cut-through switching allows for time sensitivity and lower latency applications.
- In the standard store-and-forward switching mode, cut-through switching provides more effective packet forwarding regardless of packet size. Both larger and smaller packets are transmitted before the whole packet (or frame) is received.

### Cut-through switching restrictions

- Cut-through switching is supported only between a pair of front-panel ports of the same speed or from a faster speed to a slower speed (4:1 maximum). Cut-through switching is not supported from a slower-speed to a faster-speed port.
- Cut-through switching is supported only on unicast traffic, not on multicast traffic.
- Cut-through switching is not supported on mirrored packet traffic.
- Cut-through switching is not supported on loopback, management, and CPU ports.
- Cut-through switching is not performed when a queue uses Memory Management Unit (MMU) buffers.
- Cut-through switching is not supported on encapsulated packets, such as VXLAN, ERSPAN, and Q-in-Q.
- Cut-through switching is not supported on unicast traffic that is sent from multiple source ports to the same destination port.
- Cut-through switching is not supported on an egress port that uses a back pressure mechanism with pause frames on any of its queues.
- Counters are not available to check whether a traffic flow is using cut-through or the default store-and-forward mode.

### Configure cut-through switching

The default switching mode is store-and-forward. To enable cut-through switching mode on a switch, use the `switching-mode cut-through` command.

```
sonic(config)# switching-mode cut-through
```

To disable cut-through switching mode and return the switch to the default store-and-forward mode, enter the `no switching-mode cut-through` command.

To display the configured packet switching mode, use the `show switching-mode` command.

```
sonic# show switching-mode
Current switching mode: cut-through
```

## Configure CPU polling interval

Enterprise SONiC collects average the CPU utilization data at a default interval of two minutes. You can view the collected information using the `show system cpu` command.

You can change the polling interval from the default two minutes to a different value. To change the CPU the polling interval, follow this procedure:

- Configure the following global command:

```
system resource-stats-polling-interval polling-interval
```

The range is from 120 seconds to 3600 seconds.

## Zero touch provisioning

Zero touch provisioning (ZTP) automates SONiC switch deployment and configuration without user intervention by loading a software image, running commands and scripts to bring up a switch in its wanted state.

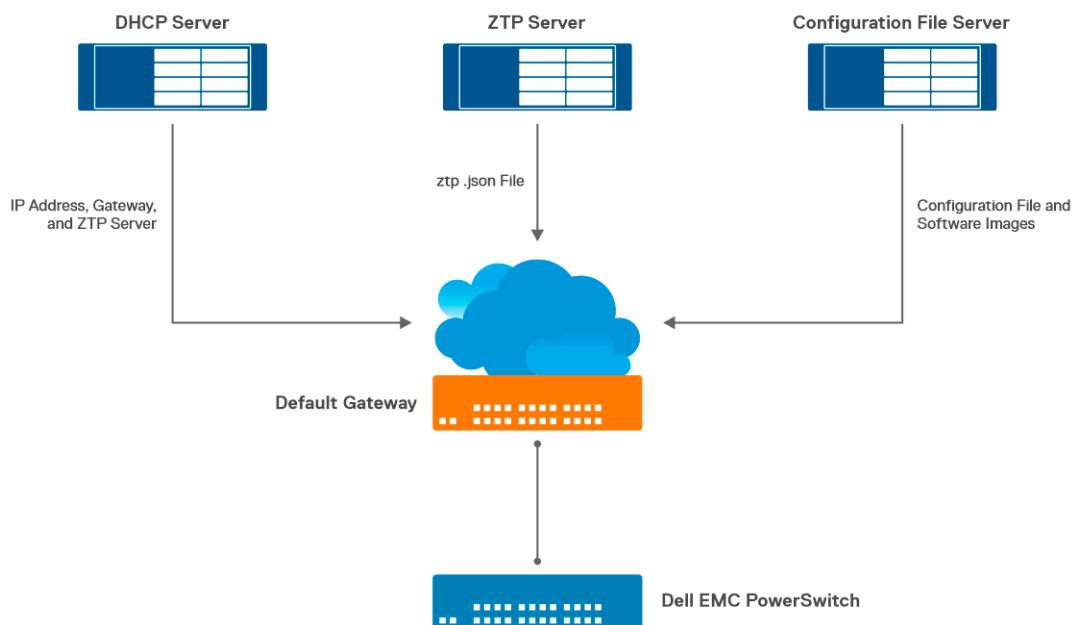
ZTP is enabled by default when you boot a switch with a factory-installed Enterprise SONiC for the first time, or when you perform an ONIE: OS Install from the ONIE boot menu.

At switch bootup, ZTP checks for a startup configuration file: `/etc/sonic/config_db.json`. If the startup configuration file is not found, ZTP creates a temporary switch configuration to perform DHCP discovery. The DHCP server sends the location of the ZTP JSON (`ztp_json_url`) or ZTP provisioning script (`ztp_provisioning_script_url`) file. If ZTP is disabled, ZTP exits and the switch boots with the factory default configuration.

Based on the DHCP information received, the switch communicates with a remote provisioning server and downloads a ZTP configuration file (`ztp.json`) or a provisioning script. ZTP performs the actions that are specified in the ZTP configuration file or runs the customized ZTP provisioning script. ZTP actions include downloading a software image, configuration files, and scripts to provision the switch, and other operations. To download these files, ZTP supports any file transfer protocol, such as HTTP, FTP, and TFTP.

This example shows how a Dell PowerSwitch uses ZTP to connect to web-based DHCP, ZTP, and configuration file servers to automate its switch configuration and firmware upgrades.

**(i) NOTE:** The DHCP, ZTP, and configuration file servers can all be on the same device.



### Topics:

- ZTP DHCP options
- ZTP JSON file
- View ZTP status
- ZTP DHCP server configuration
- Additional ZTP JSON configuration objects
- ZTP JSON file plug-ins
- ZTP JSON dynamic content
- Using ZTP plug-ins in JSON configuration objects
- ZTP provisioning using a USB drive

# ZTP DHCP options

When ZTP is enabled, the switch starts the DHCP client on all interfaces — management and front-panel ports. ZTP configures all interfaces for untagged VLAN traffic. DHCPv4 and DHCPv6 address discovery are performed on all interfaces.

The first interface that receives a valid DHCP offer is used to determine the URL for the location of the ZTP JSON file or a customized configuration script. The DHCP offer sent to a DHCP server must include:

- DHCPv4 option 67 or DHCPv6 option 59 to specify the ZTP JSON file URL
- DHCP option 239 to use a customized script specified in the ZTP provisioning URL

When the switch receives an IP address and a ZTP provisioning script or configuration from the DHCP server, it downloads and runs the script.

**(i) NOTE:** If the switch accesses the DHCP server using a front-panel port, at least one port interface must be in nonbreakout mode.

## Using a ZTP JSON file — DHCPv4 option 67 or DHCPv6 option 59

In the DHCP offer sent to the DHCP server, the DHCPv4 option 67 or DHCPv6 option 59 must include a user-defined input file in JSON format. The JSON file contains the data and logic to configure Enterprise SONiC software modules.

If a ZTP JSON file is already present on the switch, ZTP processes it and does not download a ZTP JSON from a remote server. ZTP performs configuration steps in the file which may involve multiple switches reboots. After SONiC processes the ZTP JSON file, if the startup configuration file /etc/sonic/config\_db.json is not found, ZTP restarts DHCP discovery to obtain a new ZTP JSON file and processes it.

## Using a customized script — DHCP option 239

If ZTP uses a customized script to provision a switch, specify the script URL in the ztp\_provisioning\_script\_url value in DHCP option 239.

**(i) NOTE:**

- Dell Technologies recommends that you use the ZTP JSON file option.
- If you include both DHCP option 67 for the ztp\_json\_url and DHCP option 239 for the ztp\_provisioning\_script\_url in a DHCP request, the ztp\_json\_url file is downloaded and the ztp\_provisioning\_script\_url file is ignored.
- You can specify a customized provisioning script in the ZTP JSON file as a user-defined plug-in without using the ztp\_provisioning\_script\_url option. For more information, see [ZTP JSON file plug-ins](#).

## DHCP client options

ZTP sends DHCPv4 options 61 and 77 and DHCPv6 option 15 to identify the SONiC switch from which a ZTP request originates.

**Table 16. DHCP options**

| DHCP option | Name                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 61          | dhcp-client-identifier | Uniquely identifies the switch that sends the DHCP request. Enterprise SONiC sets this value to SONiC##product-name##serial-number. The <i>product-name</i> is the ONIE EEPROM TLV 0x21 value. The <i>serial-number</i> is the ONIE EEPROM TLV 0x23 value. For example:<br><br><pre>root@sonic:/home/admin# decode-syseeprom TlvInfo Header:   Id String:      TlvInfo   Version:        1   Total Length:  193   TLV Name          Code Len Value   -----  -----  ---  -----   Product Name      0x21   9  S5248F-ON   Part Number       0x22   6  0GM4RM   Serial Number     0x23  20  TH0GM4RM-CET00076003F   Base MAC Address  0x24   6  1C:72:1D:D0:96:3F   Manufacture Date  0x25  19  07/06/2020 09:58:51   Device Version    0x26   1  1   Label Revision    0x27   3  A02   Platform Name     0x28  30  X86_64-dellemc_s5248f</pre> |

**Table 16. DHCP options (continued)**

| DHCP option         | Name                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                     |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
|---------------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|----------|---|-----|----------|--------------|------|---|-------|--|---------------------|------|---|----|--|-------------|------|---|----------|--|--------------|------|----|------------|--|-------------|------|---|---------|--|------------------|------|---|---------------------|--|--------------|------|----|------------|--|-----------------|------|----|-------------|--|--------|------|---|------------|--|
|                     |                             | <table> <tr><td>MAC Addresses</td><td>0x2A</td><td>2</td><td>256</td><td>c3538-r0</td></tr> <tr><td>Manufacturer</td><td>0x2B</td><td>5</td><td>CET00</td><td></td></tr> <tr><td>Manufacture Country</td><td>0x2C</td><td>2</td><td>TH</td><td></td></tr> <tr><td>Vendor Name</td><td>0x2D</td><td>8</td><td>Dell EMC</td><td></td></tr> <tr><td>Diag Version</td><td>0x2E</td><td>10</td><td>3.40.4.1-6</td><td></td></tr> <tr><td>Service Tag</td><td>0x2F</td><td>7</td><td>F32QY03</td><td></td></tr> <tr><td>Vendor Extension</td><td>0xFD</td><td>4</td><td>0x00 0x00 0x02 0xA2</td><td></td></tr> <tr><td>ONIE Version</td><td>0x29</td><td>10</td><td>3.40.1.1-9</td><td></td></tr> <tr><td>ONIE FW Version</td><td>0x31</td><td>11</td><td>3.40.5.1-24</td><td></td></tr> <tr><td>CRC-32</td><td>0xFE</td><td>4</td><td>0x23E7C041</td><td></td></tr> </table> | MAC Addresses       | 0x2A     | 2 | 256 | c3538-r0 | Manufacturer | 0x2B | 5 | CET00 |  | Manufacture Country | 0x2C | 2 | TH |  | Vendor Name | 0x2D | 8 | Dell EMC |  | Diag Version | 0x2E | 10 | 3.40.4.1-6 |  | Service Tag | 0x2F | 7 | F32QY03 |  | Vendor Extension | 0xFD | 4 | 0x00 0x00 0x02 0xA2 |  | ONIE Version | 0x29 | 10 | 3.40.1.1-9 |  | ONIE FW Version | 0x31 | 11 | 3.40.5.1-24 |  | CRC-32 | 0xFE | 4 | 0x23E7C041 |  |
| MAC Addresses       | 0x2A                        | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 256                 | c3538-r0 |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| Manufacturer        | 0x2B                        | 5                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | CET00               |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| Manufacture Country | 0x2C                        | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | TH                  |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| Vendor Name         | 0x2D                        | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Dell EMC            |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| Diag Version        | 0x2E                        | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 3.40.4.1-6          |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| Service Tag         | 0x2F                        | 7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | F32QY03             |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| Vendor Extension    | 0xFD                        | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x00 0x00 0x02 0xA2 |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| ONIE Version        | 0x29                        | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 3.40.1.1-9          |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| ONIE FW Version     | 0x31                        | 11                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 3.40.5.1-24         |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| CRC-32              | 0xFE                        | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x23E7C041          |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| 66                  | tftp-server                 | TFTP server address                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                     |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| 67                  | ztp_json_url                | URL used to download the ZTP JSON file. This value may specify the ZTP JSON file path on a TFTP server.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                     |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| 77                  | user-class                  | (Optional) Identifies the type or category of user or application. Enterprise SONiC sets this value to SONiC-ZTP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                     |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |
| 239                 | ztp_provisioning_script_url | Specifies the script URL which is downloaded and run by ZTP on the switch.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                     |          |   |     |          |              |      |   |       |  |                     |      |   |    |  |             |      |   |          |  |              |      |    |            |  |             |      |   |         |  |                  |      |   |                     |  |              |      |    |            |  |                 |      |    |             |  |        |      |   |            |  |

**Table 17. DHCPv6 options**

| DHCPv6 option | Name                        | Description                                                                                                       |
|---------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------|
| 15            | user-class                  | (Optional) Identifies the type or category of user or application. Enterprise SONiC sets this value to SONiC-ZTP. |
| 59            | boot-file-url               | Specifies the URL used to download the ZTP JSON file.                                                             |
| 239           | ztp_provisioning_script_url | Specifies the script URL which is downloaded and run by ZTP on the switch.                                        |

## ZTP JSON file

The ZTP JSON file supports various options to fine-tune and extend switch configuration. For more information, see the functional description in [Zero touch provisioning](#) on GitHub.

When an Enterprise SONiC switch boots for the first time, ZTP checks if there is an existing ZTP JSON file. If no JSON file exists, DHCP option 67 value is used to obtain the URL of the file. ZTP then downloads the ZTP JSON file and processes it. If the DHCP option 67 value is not included, ZTP waits indefinitely until it is provided by the DHCP server. Other switch services, including the SWSS service, continue to boot.

A ZTP JSON file consists of multiple configuration sections which are defined by the user. Each section provides data to configure a single or a set of modules on the switch.

All ZTP configuration sections are enclosed in a `ztp` object. For example, the following ZTP JSON file example in the next section has four configuration sections: `01-firmware`, `02-configdb-json`, `03-provisioningscript`, and `04-connectivity-check`, which are included as objects in the `ztp` object.

```
{
 "ztp": {
 "01-firmware": {
 "install": {
 "url": "http://192.168.1.1/sonic-image-v3.1.bin",
 "set-default": true
 },
 "reboot-on-success": true
 },
 "02-configdb-json": {
 "dynamic-url": {
 "source": {
 "url": "http://192.168.1.1/configdb-v3.1.json"
 }
 }
 }
 }
}
```

```

 "prefix": "http://192.168.1.1/",
 "identifier": "hostname",
 "suffix": "_config_db.json"
 }
}
},
"03-provisioning-script": {
 "plugin": {
 "url": "http://192.168.1.1/post_install.sh"
 },
 "reboot-on-success": true
},
"04-connectivity-check": {
 "ping-hosts": ["10.1.1.1", "yahoo.com"]
}
}
}

```

The top level `ztp` object also contains other objects and input about how to process the ZTP JSON file. In a ZTP JSON file, it is mandatory to define a `ztp` object as the first object. All other objects must appear nested in it. If the top level `ztp` object is not found, ZTP does not parse the JSON data. Also, the ZTP JSON file must be syntactically correct. If any errors are found during the parsing of the ZTP JSON file, the ZTP service exits and displays a FAILED status message.

By default, the configuration sections that are defined in the ZTP JSON file are processed in random order. It is recommended that you name the sections with a leading number, such as 01, 02, 03 in the example, so that the sections are processed in a clear numerical order.

To monitor and fine tune the ZTP processing of individual sections, you can enter various settings into each configuration section of the ZTP JSON file. For detailed information, see [ZTP JSON configuration settings](#).

### ZTP at switch startup

In the ZTP JSON file example, at the first switch bootup, the ZTP JSON file performs these configuration steps:

1. The SONiC firmware image is downloaded and installed on the switch. The switch automatically reboots to load the newly installed image.
2. The startup configuration file `config_db.json` associated with the switch is downloaded and loaded as the running configuration. The `config_db.json` file is stored with the file name `Leaf3_config_db.json` at the web root of the HTTP server with address 192.168.1.1. ZTP uniquely identifies the configuration file that is associated with each switch. In this way, multiple `config_db.json` files that are associated with different switches can be used.
3. A post-provisioning script `post_install.sh` is downloaded and run. A post-provisioning script can install and run additional applications, such as a telegraph container. The switch reboots if the `post_install.sh` script exits with a successful exit code 0.
4. A postboot connectivity check is performed by pinging hosts 10.1.1.1 and yahoo.com to verify connectivity.

### ZTP logs

ZTP generates log messages on the console about its operational status.

### Cancel ZTP in progress

To exit ZTP operation and manually configure a switch by entering CLI commands, stop the ZTP process with `no ztp enable`.

```
sonic(config)# no ztp enable
```

- A factory default configuration is created and saved as the switch startup configuration. The switch continues to operate with the loaded factory default configuration.
- The CLI session is terminated. You must log on again to access the CLI.

To re-enable ZTP, enter `ztp enable` and reboot the switch. When you re-enable ZTP, the ZTP process does not start until you reboot the switch.

```
sonic(config)# ztp enable
```

## View ZTP status

To view the status of the current ZTP session and the contents of the ZTP switch configuration, use the `show ztp status` command. The `show` output displays the configuration sections in the ZTP JSON file that are being processed and information about the last completed ZTP session.

```
sonic# show ztp-status
=====
ZTP
=====
ZTP Admin Mode : True
ZTP Service : Inactive
ZTP Status : SUCCESS
ZTP Source : dhcp-opt67 (Management0)
Runtime : 05m 31s
Timestamp : 2019-09-11 19:12:16 UTC
ZTP JSON Version : 1.0

ZTP Service is not running

01-configdb-json

Status : SUCCESS
Runtime : 02m 48s
Timestamp : 2019-09-11 19:11:55 UTC
Exit Code : 0
Ignore Result : False

02-connectivity-check

Status : SUCCESS
Runtime : 04s
Timestamp : 2019-09-11 19:12:16 UTC
Exit Code : 0
Ignore Result : False
```

- **ZTP Admin Mode** — Displays if ZTP is administratively enabled or disabled (True or False).
- **ZTP Service** — Displays ZTP status:
  - **Active Discovery** — ZTP is operational and performing DHCP discovery to learn the switch provisioning.
  - **Processing** — ZTP has discovered the switch provisioning information and is processing it .
  - **Inactive** — ZTP service is not operational.
- **ZTP Status** — Displays the current state and result of ZTP sessions:
  - **IN-PROGRESS** — ZTP is processing switch configuration information.
  - **SUCCESS** — ZTP has successfully processed the switch configuration information.
  - **FAILED** — ZTP failed to process the switch configuration information.
  - **Not Started** — ZTP has not started processing the discovered switch configuration information.
- **ZTP Source** — Displays the DHCP option and interface name from which the switch configuration information originated.
- **Runtime** — Displays the time that is taken for the ZTP process to complete from start to finish; for individual configuration sections, it indicates the time taken to process the associated configuration section.
- **Timestamp** — Displays the date/time stamp when the status field last changed.
- **ZTP JSON Version** — Version of the ZTP JSON file used for processing switch configuration information
- **Status** — Displays the current state and result of processing a ZTP JSON file:
  - **IN-PROGRESS** — The configuration section that is currently being processed
  - **SUCCESS** — The configuration section was processed successfully
  - **FAILED** — The configuration section failed to execute successfully
  - **Not Started** — ZTP has not started processing the configuration section
  - **DISABLED** — The configuration section has been marked as disabled and will not be processed
- **Exit Codes** — Displays the program exit code of the configuration section that was processed; a non-zero exit code indicates that the configuration section failed to execute successfully.

- **Ignore Results** — True indicates that the result of processing a configuration section is ignored and not used to evaluate the overall ZTP result.
- **Activity String** — Displays the current activity string, including the current action performed by ZTP, and how long it has been performing the activity.

## ZTP DHCP server configuration

For ZTP operation, configure a DHCP server in the network by adding the required ZTP option shown in **bold**. Note that multiple `ztp.json` files are used to configure different switches.

```
option domain-name "dell.org";
option domain-name-servers ns1.dell.org, ns2.dell.org;
#To install from ONIE
#option default-url "http://10.32.0.1/sonic-image-v3.1.bin";
option sonic-ztp code 67 = text;

default-lease-time 600;
max-lease-time 7200;

subnet 10.0.0.0 netmask 255.255.252.0 {
 option routers 10.0.0.253;

 group {
 host leaf1 {
 hardware ethernet 0C:04:BA:AD:DE:40;
 fixed-address 10.0.0.10;
 option sonic-ztp "http://10.0.0.1/ztp/configs/s5248f/ztp.json"; }
 host leaf2 {
 hardware ethernet 28:4F:64:8B:52:19;
 fixed-address 10.0.0.11;
 option sonic-ztp "http://10.0.0.1/ztp/configs/s5248f/ztp.json"; }
 host spine1 {
 hardware ethernet 8C:04:BA:BA:45:A1;
 fixed-address 10.0.0.12;
 option sonic-ztp "http://10.0.0.1/ztp/configs/s5232f/ztp.json"; }
 host spine2 {
 hardware ethernet 8C:04:BC:AD:2B:B2;
 fixed-address 10.0.0.13;
 option sonic-ztp "http://10.0.0.1/ztp/configs/s5232f/ztp.json"; }
 }
}
```

## Additional ZTP JSON configuration objects

To fine-tune ZTP switch configuration, enter any of the objects described in this section into a user-defined configuration section of the ZTP JSON file.

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ignore-result</b>     | Possible values: <ul style="list-style-type: none"> <li>• <code>false</code> — ZTP encountered an error while processing this configuration section; if the <code>ignore-result</code> object is not present, <code>false</code> is the default value.</li> <li>• <code>true</code> — ZTP displays the status as <code>SUCCESS</code> even if an error is encountered while processing this section.</li> </ul>                                                                    |
| <b>suspend-exit-code</b> | Displays an exit code to indicate that a configuration section is placed in suspended mode and will be retried later. For a description of each code, see <a href="#">ZTP JSON file plugins</a> . The <code>suspend-exit-code</code> object is optional in a configuration section. Possible values are non-zero positive integers. If any other value is displayed, the code is handled as if no value is specified.                                                              |
| <b>reboot-on-success</b> | Possible values: <ul style="list-style-type: none"> <li>• <code>true</code> — If the <code>status</code> result of the configuration section is <code>SUCCESS</code>, ZTP reboots the switch.</li> <li>• <code>false</code> — If ZTP successfully completes the processing of a configuration section, it moves to the next section without rebooting the switch. If the <code>reboot-on-success</code> object is not present, <code>false</code> is the default value.</li> </ul> |

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>reboot-on-failure</b>      | Possible values:                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                               | <ul style="list-style-type: none"> <li>• true — If the status result of the configuration section is FAILED, ZTP reboots the switch.</li> <li>• false — If ZTP successfully completes the processing of a configuration section, it moves to the next section without rebooting the switch. If the <code>reboot-on-failure</code> object is not present, <code>false</code> is the default value.</li> </ul>                                            |
| <b>halt-on-failure</b>        | Possible values:                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                               | <ul style="list-style-type: none"> <li>• true — If the status result of the configuration section is FAILED, ZTP stops processing the JSON file and exits. The <code>status</code> value in the top level <code>ztp</code> section is changed to FAILED. No other configuration section is processed. You must manually restart ZTP.</li> <li>• false — ZTP moves on to process the next configuration section.</li> </ul>                              |
| <b>restart-ztp-on-failure</b> | The <code>restart-ztp-on-failure</code> object applies only to the top level <code>ztp</code> section in the JSON file. Possible values:                                                                                                                                                                                                                                                                                                                |
|                               | <ul style="list-style-type: none"> <li>• true — If the status result of the configuration section is FAILED, ZTP restarts after processing all the sections in the JSON file.</li> <li>• false — ZTP exits after processing all sections in the JSON file. If the <code>restart-ztp-on-failure</code> object is not present, <code>false</code> is the default value.</li> </ul>                                                                        |
| <b>restart-ztp-no-config</b>  | The <code>restart-ztp-no-config</code> object applies only to the top level <code>ztp</code> section in the JSON file. Possible values:                                                                                                                                                                                                                                                                                                                 |
|                               | <ul style="list-style-type: none"> <li>• true — If the <code>/etc/sonic/config_db.json</code> file is not present in the JSON file, ZTP restarts after processing all sections.</li> <li>• false — If the <code>/etc/sonic/config_db.json</code> file is not present in the JSON file, ZTP exits after processing all sections. If the <code>restart-ztp-no-config</code> object is not present, <code>false</code> is the default value.</li> </ul>    |
| <b>config-fallback</b>        | The <code>config-fallback</code> object applies only to the top level <code>ztp</code> section in the JSON file. Possible values:                                                                                                                                                                                                                                                                                                                       |
|                               | <ul style="list-style-type: none"> <li>• true — If the <code>/etc/sonic/config_db.json</code> file is not present in the JSON file, ZTP installs the factory default configuration after processing all sections.</li> <li>• false — ZTP performs the action indicated by the value of the <code>restart-ztp-no-config</code> parameter. If the <code>config-fallback</code> object is not present, <code>false</code> is the default value.</li> </ul> |
| <b>ztp-json-version</b>       | Displays the version of the ZTP JSON file. Use this information to migrate ZTP JSON data between different versions of the sonic-ztp package. The <code>ztp-json-version</code> value applies only to the top level <code>ztp</code> section in the JSON file. If there is no user-entered <code>ztp-json-version</code> value, ZTP enters the version number of the current ZTP JSON file.                                                             |

## ZTP JSON file plug-ins

Each configuration section in the ZTP JSON file is processed by a corresponding plug-in, which can understand the objects in the section using a predefined logic. plug-ins are executable files, mostly scripts, which take the values that are specified in the corresponding configuration section as input.

For example, a `provisioning-script` section is processed by the predefined `provisioning-script` plug-in that is provided in the SONiC-ZTP package. The names of plug-ins that are provided in a SONiC-ZTP package must match the plug-in file name that is entered in the JSON file. Predefined plug-ins are stored in the `/usr/lib/ztp/plug-ins` directory.

### User-defined plug-ins

ZTP allows users to specify custom configuration sections and provide the corresponding executable plug-in. ZTP downloads the plug-in and uses it to process the settings specified in the configuration section. Using customized plug-ins, you can extend ZTP functionality for your network needs. To ensure better compatibility with input data, use plug-in executables that can process JSON-formatted data.

This sample section of a ZTP JSON file is used to configure SNMP communities on a switch. The `plugin` setting defines the usage of user-defined plug-in. In this example, the user-provided `my-snmp.py` file is downloaded using the URL specified by the `plugin.url.source` field. The plug-in is copied locally as the `/var/run/ztp/plugins/my-snmp` file on the switch and run by ZTP. If a `plugin.url.destination` path is not specified, the downloaded plug-in is saved in the `/var/lib/ztp/sections/JSON-file-section-name/plugin` directory.

```
"snmp": {
 "ignore-result": false,
```

```

"plugin": {
 "url": {
 "source": "http://192.168.1.1:8080/my-snmp.py",
 "destination": "/var/run/ztp/plugins/my-snmp"
 }
},
"community-ro": [
 "public",
 "local"
],
"community-rw": [
 "private"
]
}

```

User-defined plug-ins are downloaded only once during a ZTP service. If the destination file exists locally, the plug-in is not downloaded again. Dell Technologies recommends that you do not use the `plugin.url.destination` setting and instead allow ZTP to download a plug-in file to temporary storage. The temporary storage is cleared when a new ZTP session starts and is guaranteed not to conflict with plug-ins used in other configuration sections.

#### Plug-in exit code

Both predefined and user-defined plug-ins are run as a ZTP subprocess. ZTP uses the exit code that is generated after a plug-in is run to take the next processing step. The value of the `suspend-exit-code` object is entered in each configuration section.

**Table 18. Plug-in exit codes**

| Exit code                                                | ZTP action                                                                                                                                                |
|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                                                        | The plug-in configuration operation completed successfully. The <code>status</code> setting in the configuration section is set to <code>SUCCESS</code> . |
| <code>suspend-exit-code</code> value                     | The plug-in operation exited after partial completion. The <code>status</code> setting in the configuration section is set to <code>SUSPEND</code> .      |
| Greater than 0 or <code>!=suspend-exit-code</code> value | An error occurred when running the plug-in. The <code>status</code> setting in the configuration section is set to <code>FAILED</code> .                  |

If a plug-in uses the `suspend-exit-code` value to suspend ZTP operation, when ZTP tries to run the plug-in again, ensure that appropriate logic is implemented in the plug-in so that it restarts from where it stopped in the previous ZTP session. If the `suspend-exit-code` value is not specified, the suspend and resume feature for the plug-in is not supported.

## ZTP JSON dynamic content

When processing a ZTP JSON file, a switch may request a configuration file from a specified URL location. The server locates the file and downloads it on the switch. The server identifies the file in either of two ways:

- HTTP headers on the server side
- Dynamic URLs on the client side

#### HTTP headers

To request a configuration file, ZTP sends an HTTP/HTTPS request to a server and includes this switch information in the HTTP header.

**Table 19. HTTP headers**

| HTTP header field | Enterprise SONiC switch               | Example             |
|-------------------|---------------------------------------|---------------------|
| User-Agent        | SONiC-ZTP/0.1                         | ---                 |
| Product-Name      | <i>switch-model</i>                   | E1031               |
| Serial-Number     | Factory-assigned <i>serial-number</i> | E1031B2F035A17GD020 |
| Service-Tag       | Factory-assigned <i>service-tag</i>   | 75T00Q2             |
| Base-MAC-Address  | Factory-assigned <i>mac-address</i>   | 00:E0:EC:38:50:FB   |

**Table 19. HTTP headers (continued)**

| <b>HTTP header field</b> | <b>Enterprise SONiC switch</b>                         | <b>Example</b>             |
|--------------------------|--------------------------------------------------------|----------------------------|
| Enterprise-SONiC-Version | <i>version-number</i> displayed in show version output | 3.1.0_RC4-Enterprise_Base' |

**Dynamic URL**

In a ZTP JSON file, you can enter a dynamic URL as a nested object in a configuration section:

```
"02-configdb-json": {
 "dynamic-url": {
 "source": {
 "prefix": "http://192.168.1.1/",
 "identifier": "hostname",
 "suffix": "_config_db.json"
 }
 }
}
```

A dynamic URL constructs the URL path for a configuration file dynamically at runtime. Using a dynamic URL allows the switch to request the appropriate file when it is needed, instead of relying on the server to interpret HTTP headers to locate the file. In addition, because other transport protocols, such as SCP, TFTP, and FTP, may also be supported, the use of HTTP header information may not be valid.

In a JSON file, enter the `dynamic-url.source` field using three subobjects, whose values are used to construct the URL at runtime. The `identifier` value is mandatory. The `prefix` and `suffix` subobjects are optional.

|                   |                                                                                                                                                                                                                                |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>prefix</b>     | Specifies a static string which forms the leading part of the URL, such as the transfer protocol and server address ( <code>http://192.168.1.1:8080/</code> ).                                                                 |
| <b>identifier</b> | Specifies the runtime logic used to generate the filename to be downloaded. For example, you can enter the hostname of the switch, the serial number, the product name, or a script name as the <code>identifier</code> value. |
| <b>suffix</b>     | Specifies the filename extension to be added.                                                                                                                                                                                  |

The prefix, identifier, and suffix are concatenated to form the URL, which is specific to the requesting switch. Some sample `dynamic-url.source` entries include:

```
"dynamic-url": {
 "source": {
 "prefix": "http://192.168.1.1:8080/configs/",
 "identifier": "hostname-fqdn",
 "suffix": "_config_db.json"
 },
 "destination": "/etc/sonic/config_db.json"
}

"dynamic-url": {
 "source": {
 "prefix": "http://192.168.1.1:8080/configs/",
 "identifier": {
 "url": "http://192.168.1.1:8080/config_filename_eval.sh"
 },
 "suffix": "_config_db.json"
 },
 "destination": "/etc/sonic/config_db.json"
}
```

**Table 20. Supported dynamic-url subobjects**

| <b>Nested url object</b>   | <b>Description</b>       | <b>Supported values</b>                           | <b>Default value</b> |
|----------------------------|--------------------------|---------------------------------------------------|----------------------|
| <code>source.prefix</code> | First part of URL string | Syntactically valid URL                           | Null string          |
| <code>source.suffix</code> | Last part of URL string  | Supported characters in a syntactically valid URL | Null string          |

**Table 20. Supported dynamic-url subobjects (continued)**

| <b>Nested url object</b> | <b>Description</b>                                                        | <b>Supported values</b>                                                                                                                                                                                                                                                                                         | <b>Default value</b>    |
|--------------------------|---------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| source.identifier        | Runtime-generated string that uniquely identifies a switch                | Any of the following: <ul style="list-style-type: none"><li>● <code>hostname</code></li><li>● <code>hostname-fqdn</code></li><li>● <code>serial-number</code></li><li>● <code>service--tag</code></li><li>● <code>product-name</code></li><li>● <code>sonic-version</code></li><li>● <code>url</code></li></ul> | N/A                     |
| destination              | Specifies the destination path on the switch, including the file name     | UNIX file name                                                                                                                                                                                                                                                                                                  | Resolved by the plug-in |
| secure                   | Enables secure mode to avoid security checks when HTTPS transport is used | true or false                                                                                                                                                                                                                                                                                                   | true                    |
| curl-arguments           | Curl arguments used to download a file from the specified URL             | See the <a href="#">curl manual</a> .                                                                                                                                                                                                                                                                           | Null string             |
| encrypted                | Indicates that the file being downloaded is in encrypted format           | See Encryption.                                                                                                                                                                                                                                                                                                 | No encryption is used.  |
| include-http-headers     | Enables/disables switch information to be sent in HTTP headers            | true or false                                                                                                                                                                                                                                                                                                   | true                    |
| timeout                  | Maximum number of seconds allowed for curl to establish a connection      | Valid non-zero integer                                                                                                                                                                                                                                                                                          | 30 seconds              |

## Using ZTP plug-ins in JSON configuration objects

This section describes the supported user-defined plug-ins and the corresponding configuration sections to create in the JSON file.

The user-defined objects that are supported in configuration sections in a ZTP JSON file are:

- `url`
- `plugin`
- `configdb-json`
- `firmware`
- `connectivity-check`
- `snmp`
- `download`

### **url**

The `url` object specifies a file that can be downloaded from a remote location. The `source` subobject defines the source URL and is mandatory. All other `url` subobjects are optional. The `destination` subobject defines the complete path and file name on the switch where the file is copied. When you specify a destination, all subdirectories in the path are also created. By default, the destination and filename are determined by the plug-in:

```
"url": {
 "source": "http://192.168.1.1:8080/spine01_config_db.json",
 "destination": "/etc/sonic/config_db.json",
 "secure": false
}
```

The curl application is used to download the file. For supported protocols, see the [curl manual](#).

- The source URL defined in the `url` object is a static string. When this object is processed, the same file is downloaded by all switches that use the same ZTP JSON file. You may sometimes want to download a file, such as `config_db.json`,

with unique data for certain switches. In this case, you would need to create different config\_db.json files with unique configurations.

- If source is the only object specified in the url object, you can enter the url object in shorthand notation:

```
"url": "http://192.168.1.1:8080/spine01_config_db.json"
```

**Table 21. Supported url subobjects**

| url subobject        | Description                                                                | Supported values            | Default value           |
|----------------------|----------------------------------------------------------------------------|-----------------------------|-------------------------|
| source               | Specifies the URL from where the file is downloaded.                       | Any syntactically valid URL | N/A                     |
| destination          | Specifies the destination path on the switch string and the file name.     | UNIX file name              | Resolved by the plug-in |
| secure               | Enables secure mode to avoid security checks when HTTPS transport is used. | true or false               | true                    |
| curl-arguments       | Curl arguments used to download a file from the specified source URL.      | See the curl manual.        | Null string             |
| encrypted            | Indicates that the file being downloaded is in encrypted format.           | See Encryption.             | No encryption is used   |
| include-http-headers | Enables/disables switch information to be sent in HTTP headers.            | true or false               | true                    |
| timeout              | Maximum number of seconds allowed for curl to establish a connection.      | Valid non-zero integer      | 30 seconds              |

## plugin

The plugin object specifies the user-defined plug-in to be used to process a configuration section. For example, you can configure the config-db-json plug-in that is provided in the Enterprise SONiC ZTP package to be used in the initial-config section to download and apply the initial switch configuration:

```
"url": {
 "source": "http://192.168.1.1:8080/spine01_config_db.json",
 "destination": "/etc/sonic/config_db.json",
 "secure": false
}
"initial-config": {
 "plugin": {
 "name": "config-db-json"
 },
 "url": {
 "source": "http://192.168.1.1:8080/spine01_first_boot_config.json",
 "destination": "/etc/sonic/config_db.json",
 "secure": false
 }
}
```

- The dynamic-url takes precedence over the url object if url and dynamic-url definitions are defined.
- You can enter the plugin object in shorthand notation by using a "plugin": *name* value; for example:

```
"plugin": "configdb-json"
```

**Table 22. Supported plugin subobjects**

| plugin subobject | Description                                             | Supported values                               | Default value            |
|------------------|---------------------------------------------------------|------------------------------------------------|--------------------------|
| url              | Specifies the URL from where the plug-in is downloaded. | See <b>url</b> in this section.                | Name of enclosing object |
| dynamic-url      | Specifies the URL from where the plug-in is downloaded. | See <b>dynamic-url object</b> in this section. | Name of enclosing object |

**Table 22. Supported plugin subobjects (continued)**

| <b>plugin subobject</b> | <b>Description</b>                                                                                                                                                                                                     | <b>Supported values</b> | <b>Default value</b>     |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|--------------------------|
| name                    | Specifies a predefined plug-in that is available in the Enterprise SONiC bundle.                                                                                                                                       | Predefined plug-ins     | Name of enclosing object |
| shell                   | Specifies if the plug-in is run through the shell.                                                                                                                                                                     | true or false           | false                    |
| ignore-section-data     | Specifies if the data read section in the configuration section of the plug-in is passed as the first argument with the plugin command.                                                                                | true or false           | false                    |
| args                    | Defines the argument string that is passed as an argument with the plugin command. If the plugin command expects only the argument string as part of the command, you must set the ignore-section-data object to true. | Valid command arguments | Null string              |
| pty                     | Specifies if the plug-in is run using a pseudo terminal.                                                                                                                                                               | true or false           | false                    |
| vrf                     | Specifies the VRF in which the plug-in is run. To run the plug-in enter the command: <code>cgexec -g 13mdev:vrf-name</code>                                                                                            | mgmt                    | N/A                      |

### configdb-json

The configdb-json plug-in downloads the config\_db.json file and applies the configuration. The switch reloads the configuration:

```
"configdb-json": {
 "url": {
 "source": "http://192.168.1.1:8080/spine01_config_db.json",
 "destination": "/etc/sonic/config_db.json"
 }
}
```

- You must specify either a `url` or a `dynamic-url` subobject in the `configdb-json` configuration section in the JSON file.
- When the configdb-json plug-in runs, the DHCP address assigned during ZTP discovery is released. Be sure to configure an interface IP address assignment in the downloaded config\_db.json file.
- If you use different config\_db.json files for multiple switches, the files may have the same DEVICE\_METADATA parameters even if the PORT, INTERFACE, and other configurations are unique and accurate. As a result, the same MAC address, HW SKU, and platform string are configured on each switch. To resolve the inaccuracies in the config\_db.json-applied configurations, use the configdb-json plug-in to overwrite the MAC, HW SKU, and platform fields in the DEVICE\_METADATA local host entry in the JSON configuration file that is downloaded using ZTP. The corresponding accurate values are read from the current running configuration and used in the user-provided config\_db.json file. If the DEVICE\_METADATA entry is missing in the config\_db.json file, the DEVICE\_METADATA values in the running configuration are entered in the file.

**Table 23. Supported configdb-json subobjects**

| <b>configdb-json subobject</b> | <b>Description</b>                                                  | <b>Supported values</b>                              | <b>Default value</b> |
|--------------------------------|---------------------------------------------------------------------|------------------------------------------------------|----------------------|
| url                            | Specifies the URL from where the config_db.json file is downloaded. | See <code>url</code> in this section.                | N/A                  |
| dynamic-url                    | Specifies the URL from where the config_db.json file is downloaded. | See <code>dynamic-url object</code> in this section. | N/A                  |

**Table 23. Supported configdb-json subobjects (continued)**

| configdb-json subobject | Description                                                                                                                                                                                                                                                                                                                                                                                                                                           | Supported values                                                                                                | Default value |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|---------------|
| clear-config            | Specifies if the current configuration is cleared before loading the content of the config_db.json file. If set to true, the config reload command is run. If set to false, the config load command is run. <b>Note:</b> When clear-config flag is set to false, if the ZTP profile is active, the factory-default configuration is loaded first. Afterwards, the user-provided configuration is applied on top of the factory-default configuration. | true or false                                                                                                   | true          |
| save-config             | Runs the config save command loading the content of the config_db.json file.                                                                                                                                                                                                                                                                                                                                                                          | true or false                                                                                                   | true          |
| frr-config              | Use this option to download the FRR configuration file that is stored at /etc/sonic/frr/frr.conf. Enter the URL of the FRR configuration file a URL string or a dynamic-url object.                                                                                                                                                                                                                                                                   | URL string, url object, or dynamic-url object. See the url and dynamic-url object descriptions in this section. | N/A           |
| default-config          | Use this option to load the factory-default configuration. <b>Note:</b> The loaded factory-default configuration is not automatically saved to /etc/sonic/config_db.json.                                                                                                                                                                                                                                                                             | true or false                                                                                                   | false         |
| config-reload           | Use this option to run the config reload command and load the startup-config file, , /etc/sonic/config_db.json, in the Config_DB redis database, and restart all docker containers.                                                                                                                                                                                                                                                                   | true or false                                                                                                   | false         |

## firmware

Use the firmware plug-in to manage the Enterprise SONiC image on the switch. The firmware plug-in can install, remove, and boot selected images. It uses the sonic\_installer utility to perform these activities.

For example, to install an image and reload the switch with the new image:

```
"firmware": {
 "install": {
 "url": "http://192.168.1.1:8080/sonic-image-v3.1.bin",
 "set-default": true
 },
 "reboot-on-success": true
}
```

To uninstall an image on the switch:

```
"firmware": {
 "remove": {
 "image": "SONiC-OS-brcm_xlr_gts.0-dirty-20190304.154831"
 },
 "reboot-on-success": true
}
```

To install a new image only if it satisfies the pre-install verify check script provided by user:

```
"firmware": {
 "install": {
 "url": "http://192.168.1.1:8080/sonic-image-v3.1.bin",
 "pre-check": {
 "url": {
 "source": "http://192.168.1.1:8080/firmware_check.sh",
 "destination": "/tmp/firmware_check.sh"
 }
 },
 "set-default": true
 },
 "reboot-on-success": true
}
```

- Use the `pre-check` subobject to specify a user-provided script to run. If the result of the script is successful, the `install` or `remove` action is performed. The `pre-check` object takes a URL value.
- If a `remove` and an `install` URL are both specified, the `remove` firmware operation is performed first, followed by the `install` firmware.

**Table 24. Supported firmware subobjects**

| <b>firmware subobject</b> | <b>Description</b>                                                                                                                                                                                      | <b>Supported values</b>               | <b>Default value</b> |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|----------------------|
| install                   | Specifies the URL used to install an image.                                                                                                                                                             | See <code>url</code> in this section. | N/A                  |
| url dynamic-url           | Specifies the URL from where the firmware image is downloaded.                                                                                                                                          | URL or dynamic URL                    | N/A                  |
| version                   | Specifies the version number of the Enterprise SONiC image to be installed. This value is optional.                                                                                                     | Enterprise SONiC version number       | N/A                  |
| set-default               | Specifies that the firmware image being installed is used as the default image to load at switch bootup.                                                                                                | true or false                         | true                 |
| set-next-boot             | Specifies that the firmware image being installed is selected as the image to load only once at the next switch reboot.                                                                                 | true or false                         | false                |
| skip-reboot               | Specifies if a switch reboot operation is performed immediately after installing a new switch firmware image.                                                                                           | true or false                         | false                |
| pre-check                 | Specifies the URL of a user-provided script to be run before installing the downloaded firmware image. The firmware installation is performed only if the result of the pre-check script is successful. | URL or dynamic URL                    | N/A                  |
| remove                    | Uninstalls an existing image on the switch.                                                                                                                                                             | N/A                                   | N/A                  |
| version                   | Specifies the version of the Enterprise SONiC image to be removed. This value is mandatory.                                                                                                             | Enterprise SONiC version number       | N/A                  |
| pre-check                 | Specifies the URL of a user-provided script to be run before removing the specified firmware image version. The firmware removal is performed only if the                                               | URL or dynamic URL                    | N/A                  |

**Table 24. Supported firmware subobjects (continued)**

| <b>firmware subobject</b> | <b>Description</b>                                                                                                                                                                                           | <b>Supported values</b>                            | <b>Default value</b> |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|----------------------|
|                           | result of the pre-check script is successful.                                                                                                                                                                |                                                    |                      |
| upgrade-docker            | Installs a docker image on the switch.                                                                                                                                                                       | N/A                                                | N/A                  |
| url dynamic-url           | Specifies the URL from where the docker image file is downloaded.                                                                                                                                            | URL or dynamic URL                                 | N/A                  |
| container-name            | Name of the docker image being upgraded                                                                                                                                                                      | Supported docker container name; for example, swss | N/A                  |
| cleanup-image             | Deletes an older docker image while installing a new docker image.                                                                                                                                           | true or false                                      | false                |
| enforce-check             | Enforces the pending task check for a docker upgrade.                                                                                                                                                        | true or false                                      | false                |
| tag                       | Specifies a tag for the newly installed docker image.                                                                                                                                                        | Valid text string                                  | Null string          |
| pre-check                 | Specifies the URL of a user-provided script to be run before installing the specified docker container. The docker image installation is performed only if the result of the pre-check script is successful. | URL or dynamic URL                                 | N/A                  |

#### **connectivity-check**

Use the `connectivity-check` plug-in to ping a remote device and verify if the switch can reach the remote host. Using this plug-in, you can ping multiple devices. The connectivity check fails even if ping to one of the specified host fails.

```
"connectivity-check" : {
 "ping-hosts" : ["192.168.1.1", "172.10.1.1"]
}
```

**Table 25. Supported connectivity-check subobjects**

| <b>connectivity-check subobject</b> | <b>Description</b>                                                                                                                                                                                            | <b>Supported values</b> | <b>Default value</b> |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|----------------------|
| ping-hosts                          | Specifies the IPv4 hosts to ping.                                                                                                                                                                             | N/A                     | N/A                  |
| ping6-hosts                         | Specifies the IPv6 hosts to ping.                                                                                                                                                                             | N/A                     | N/A                  |
| retry-interval                      | Specifies a timeout (in seconds) before retrying to ping a remote device.                                                                                                                                     | Valid non-zero number   | 5 seconds            |
| retry-count                         | Stops pinging a device, moves to the next device in the list and starts to ping it.                                                                                                                           | Valid non-zero number   | 12 attempts          |
| ping-count                          | Stops pinging a device after sending the specified number of ECHO_REQUEST packets. If you specify a deadline timeout, the ping waits to receive the specified number of ECHO_REPLY packets before timing out. | Valid non-zero number   | Send 3 packets       |
| deadline                            | Specifies a timeout (in seconds) when a ping stops regardless of how many packets are sent or                                                                                                                 | Valid non-zero number   | N/A                  |

**Table 25. Supported connectivity-check subobjects (continued)**

| <b>connectivity-check subobject</b> | <b>Description</b>                                                                                                                                                                                                                                            | <b>Supported values</b> | <b>Default value</b> |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|----------------------|
|                                     | received. The ping does not stop after the specified number of ping-count packets are sent. The ping stops only after the deadline timeout or when the specified number of ping-count reply packets are received or when a network error message is received. |                         |                      |
| timeout                             | Specifies the time (in seconds) to wait for a response from a remote device before using the retry-interval and retry-count settings.                                                                                                                         | Valid non-zero number   | N/A                  |

**download**

Use the download plug-in to download the list of files defined in the `url` and `dynamic-url` subobjects in the `plugin` object. Using the download plug-in, you can download various configuration files and scripts to be used in later sections in the ZTP JSON file. For example:

```
"01-download": {
 "files": [
 {
 "dynamic-url": {
 "source": {
 "prefix": "http://192.168.1.1/",
 "identifier": "hostname",
 "suffix": "_config_db.json"
 },
 "destination": "/home/admin/config_db.json"
 }
 },
 {
 "dynamic-url": {
 "source": {
 "prefix": "http://192.168.1.1/",
 "identifier": "hostname",
 "suffix": "_frr.conf"
 },
 "destination": "/etc/sonic/frr/frr.conf"
 }
 },
 {
 "url": {
 "source": "http://192.168.1.1/post-install.sh",
 "destination": "/home/admin/post-install.sh"
 }
 }
]
}
```

**Table 26. Supported download subobjects**

| <b>connectivity-check subobject</b> | <b>Description</b>                                                  | <b>Supported values</b>                                           | <b>Default value</b> |
|-------------------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------|----------------------|
| files                               | Specifies a list of files to be downloaded by the download plug-in. | JSON list of <code>url</code> or <code>dynamic-url</code> objects | N/A                  |

# ZTP provisioning using a USB drive

Zero touch provisioning supports the use of USB storage media to automate SONiC switch deployment and configuration.

When a SONiC switch boots up, if there is no startup configuration file `/etc/sonic/config_db.json` detected, the ZTP service initiates a discovery process. ZTP sets up in-band interfaces and starts DHCP discovery on in-band and out-of-band interfaces. As part of the ZTP discovery process, ZTP also scans the USB interface to identify if any storage media is connected to the switch.

- If detected, the USB storage media is mounted and its contents are searched for the presence of a ZTP JSON file. ZTP searches for the `ztp.json` file in the root directory of the attached USB storage media.
- If found, ZTP processes the `ztp.json` file. The configuration sections listed in the ZTP JSON file are processed and executed — see [ZTP JSON file](#).

**i | NOTE:** When the ZTP service is active, installed USB media is always kept mounted and ready to be used during all the steps for complete switch provisioning. The USB media is unmounted only when the ZTP service exits.

In addition to the `ztp.json` file, you can also store initial configuration files, provisioning scripts, user-defined plugin scripts, and firmware images on USB storage media. To access the files, use the file path `/media/usb0/filename` — see [Using USB storage media](#) for more information. If you use the url ([Using ZTP plug-ins in JSON configuration objects](#)) or dynamic-url ([ZTP JSON dynamic content](#)) object, specify the path `file:///media/usb0/filename` to access the files stored on the USB storage media.

Before processing the provisioning information on USB media, ZTP initializes the switch configuration using the default ZTP configuration. The switch can then establish connectivity through in-band or out-of-band interfaces during USB provisioning. In this way, you can use a combination of provisioning data on the USB media and from remote servers over the network to fully provision the switch. However, when a ZTP JSON file is read from USB media and processed, the ZTP JSON file cannot be obtained in other ways, such as using DHCP options. During the ZTP discovery process, a ZTP JSON file found on USB media is given priority over the ZTP JSON URL received using DHCP options.

If a switch has multiple USB slots or if the inserted USB media has multiple partitions, all discovered USB device nodes are automatically mounted in the order of their detected device numbers. The file path where a USB device is mounted is automatically assigned in the format, `/media/usb#`, where `#` is between 0 and 7 for the number of an installed USB device. You can store provisioning information in any of the attached USB devices. ZTP searches all USB devices in order from 0 to 7 and uses the first discovered provisioning information.

**i | NOTE:** Dell Technologies strongly recommends that you always use a USB storage media with a single partition. In addition, install the USB drive in the same USB slot on the switch to create a predictable directory path where USB storage media can be accessed (for example, `/media/usb0`). All configuration scripts can refer to this fixed path.

**i | NOTE:** When a ZTP session using USB storage is in progress, if you remove or insert USB media, the storage changes may not be identified and may result in configuration failures. However, any changes to attached USB devices during the ZTP discovery stage are well-received and can result in identifying the newly provided provisioning information to complete the configuration of the switch. Dell Technologies strongly recommends that attached USB storage is set up with all required provisioning information before the ZTP session starts.

## Example: ZTP JSON file for USB storage media

The following ZTP JSON file shows an example of how to provision a switch using files stored on USB storage media. For more information about how to create a ZTP JSON file, see [ZTP JSON file](#).

```
{
 "ztp": {
 "01-firmware": {
 "install": {
 "url": "file:///media/usb0/onie-installer.bin",
 },
 },
 "02-configdb-json": {
 "url": "file:///media/usb0/config_db.json"
 },
 "03-provisioning-script": {
 "plugin": {
 "url": "file:///media/usb0/install.sh"
 },
 }
 }
}
```

```
}
```

#### Example: USB file contents

```
root@sonic:/# tree /media/usb0/
/media/usb0/
├── config_db.json
├── install.sh
└── onie-installer.bin
└── ztp.json
```

When ZTP is performed using USB provisioning files, the ZTP source is displayed as `usb`. The ZTP JSON file is also displayed. For example:

```
root@sonic:/home/admin# ztp status
ZTP Admin Mode : True
ZTP Service : Inactive
ZTP Status : SUCCESS
ZTP Source : usb(/media/usb0/ztp.json)
ZTP Runtime : 09m 06s
ZTP Timestamp : 2022-03-01 02:03:36 UTC
ZTP Service is not running

01-firmware: SUCCESS
02-configdb-json: SUCCESS
03-provisioning-script: SUCCESS
```

#### Additional documentation

For more information about how to create a `ztp.json` file, see [Zero Touch Provisioning \(ZTP\)](#).

 **NOTE:** When using information in the `ztp.md` file for USB-based ZTP provisioning, be sure to replace all occurrences of "`http://ip-address/`" with "`file:///media/usb0/`".

# Interfaces

|                                    |                                                                                                                                                                                                                                      |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Interface configuration</b>     | Configures Ethernet, VLAN, loopback, port channel, and the management interface using the Management Framework CLI (see <a href="#">Basic interface configuration</a> ).                                                             |
| <b>Access and trunk interfaces</b> | Configures an Ethernet interface to operate either as a trunk (tagged traffic) or access (untagged traffic) port (see <a href="#">Configure access and trunk interfaces</a> ).                                                       |
| <b>Port channels</b>               | Bundles multiple physical interfaces into a single logical interface — a link aggregation group (LAG) or port channel — for redundancy, increased bandwidth, and better traffic load-balancing (see <a href="#">Port channels</a> ). |
| <b>VLANs</b>                       | Connects devices for common communication needs, independent of physical location (see <a href="#">VLANs</a> ).                                                                                                                      |
| <b>Port groups</b>                 | Port-groups allow you to apply common configurations on a set of ports according to their hardware characteristics (see <a href="#">Port groups</a> ).                                                                               |
| <b>Port breakouts</b>              | Using a supported breakout cable, you can split a 40GE QSFP+, 100GE QSFP28, or 400GE QSFP56-DD Ethernet port into separate breakout interfaces (see <a href="#">Port breakouts</a> ).                                                |

## Topics:

- Interface configuration mode
- Basic interface configuration
- Configure access and trunk interfaces
- Interface ranges
- Forward error correction
- Port groups
- Port profiles
- Port breakouts
- Port channels
- VLANs
- Q-in-Q VLAN tunneling
- VLAN translation
- Layer 3 subinterfaces
- Show transceivers
- Media-based port autoconfiguration
- High-power optics

## Interface configuration mode

Use the Management Framework CLI to configure Ethernet, VLAN, loopback, port channel, and the management interface. Enter configuration commands in Interface Configuration mode.

### Ethernet interfaces

To configure an Ethernet interface, enter the port number of the interface to be configured; for example:

```
sonic(config)# interface Eth1/4
sonic(conf-if-Eth1/4) #
```

### Management interface

To configure the management interface, enter `interface Management 0`.

```
sonic(config)# interface Management 0
sonic(conf-if-Management0) #
```

## Loopback interfaces

To configure a loopback interface, enter `interface Loopback number` (0 to 16383).

```
sonic(config)# interface Loopback 10
sonic(conf-if-lo10) #
```

## VLAN interfaces

To configure a VLAN interface, enter `interface Vlan vlan-id` (1 to 4094).

```
sonic(config)# interface Vlan 10
sonic(conf-if-Vlan10) #
```

## Port-channel interfaces

To configure a port-channel interface, enter `interface PortChannel number` (1 to 256).

```
sonic(config)# interface PortChannel 10
sonic(conf-if-po10) #
```

# Basic interface configuration

Use the commands to configure basic settings on Ethernet, Management, loopback, VLAN, and port channel interfaces. By default, an Ethernet or port channel interface cannot receive or transmit L2 or L3 traffic unless it is configured as a VLAN member or assigned an IP address.

**(i) NOTE:** On an Enterprise SONiC switch, ports have a Link and an Activity LED. Sometimes the Port and Activity LEDs are combined in a single Link/Activity LED. A solid green Link LED means that a link is present and is operational. A flashing green Activity LED means that the port is sending or receiving data. If the Link LED is off, there is no link; if the Activity LED is off, there is no port activity.

**Autonegotiation** — (Management interface) Set a Management interface to autonegotiate speed with a connected device. Autonegotiation is disabled by default. Both sides of a link must have autonegotiation enabled or disabled for the link to come up.

- `autoneg on` — Enables interface speed autonegotiation.
- `autoneg off` — Disables interface speed autonegotiation.
- `no autoneg` — Resets interface speed autonegotiation to the default: `autoneg off`.

**Autonegotiation** — (Ethernet interfaces) An Ethernet interface autonegotiates speed with a connected device so that both transmit data at the highest set of common capabilities. Both sides of a link must have autonegotiation enabled or disabled for the link to come up. When enabled, autonegotiation automatically configures interface speed over twisted-pair and copper links. Autonegotiation is enabled by default only on copper RJ45 ports.

- `no speed auto` — Disables interface speed autonegotiation.
- `speed auto` — Re-enables interface speed autonegotiation.

**(i) NOTE:** When a link is established, autonegotiation does not guarantee a compatible mode of operation. In SFP, QSFP, and QSFP-DD links, autonegotiation automatically has link training enabled to ensure optimal hardware tuning between connected devices. When autonegotiation is disabled, configure standalone link training on the interface to activate dynamic hardware tuning between connected devices — see **Link training** in this section.

**(i) NOTE:** Autonegotiation is enabled by default on copper RJ45 ports. Dell Technologies recommends that you do not disable autonegotiation on RJ45 ports.

**(i) NOTE:** Starting in Release 4.1.0, autonegotiation is supported on the SFP and QSFP ports of N3200- and E3200-series switches.

**Description** — (Ethernet, Management, VLAN, port channel, and loopback interfaces) Enter a text description of an interface. Use the `show interface` command to view the descriptions configured for each interface.

- `description string` — Enter up to 240 characters. Special characters are allowed. To use spaces between characters, enclose the entire description in quotation marks; for example, “text description”. The text string that you enter overwrites any previously configured text string.

**IP address** — Configure an IPv4 address on an interface

- `ip address ip-address/mask` — Enter the IP address in dotted decimal format *A.B.C.D*. The `no ip address ip-address` command removes the IP address on the interface.

**IPv6 address** — Configure an IPv6 address on an interface.

- `ipv6 address ipv6-address/prefix-length` — Enter a full 128-bit IPv6 address with the network prefix length, including the 64-bit interface identifier. You can configure multiple IPv6 addresses on an interface. The `no ipv6 address ipv6-address` command removes an IPv6 address on the interface.
- To configure an IPv6 address besides the link-local address, use the `ipv6 address ipv6-address/prefix-length` command and specify the complete 128-bit IPv6 address.

**MTU** — Configure the maximum transmission unit (MTU) frame size for L2 or L3 traffic on an interface.

- `mtu value` — Enter the maximum frame size in bytes (1312 to 9216; default 9100). Enter `no mtu` to reset the default value.

**Shutdown** — Disable an interface for incoming and outgoing traffic. To re-enable an interface, use the `no shutdown` command. When you shut down a VLAN, the L3 functions within the VLAN are disabled, and L2 traffic continues to flow.

**Speed** — (Management interface) Set the speed at which a Management interface transmits and receives data.

- `speed {10 | 100 | 1000 | auto}` — Enter the interface speed in Mbps. 1000 Mbps is 1 Gbps. The default is 1000, or 1 Gbps. Enter `auto` to configure autonegotiated speed on a Management interface.

**Speed** — (Ethernet interfaces) Set the advertised speed(s) at which an Ethernet interface transmits and receives data. When you reconfigure interface speed, enter `auto` to keep autonegotiation enabled. If you do not enter `auto`, autonegotiation is disabled.

- `speed {10 | 100 | 1000 | 2500 | 5000 | 10000 | 20000 | 25000 | 40000 | 50000 | 100000 | 200000 | 400000}` — Enter the interface speed(s) in Mbps. 1000 Mbps is 1 Gbps. All speeds are for full-duplex transmission. The default is 1000, or 1 Gbps.

Separate advertised speeds with a comma; for example: `speed 25000,10000,1000`. Enter `speed auto` to enable autonegotiation and advertise all supported speeds. When you enable autonegotiation, you can specify the speeds to advertise to connecting devices; for example: `speed auto 25000`. Dell Technologies recommends that you advertise the native speed of the cable used in the port.

**i** **NOTE:** If autonegotiation is off, you must set the interface speed.

**i** **NOTE:** Autonegotiation and link training are not supported on the 10G SFP+ ports of Z9664F-ON, Z9432F-ON, Z9332F-ON, Z9264F-ON, S5448F-ON, S5232F-ON, N3248TE-ON, and E3248P-ON switches.

**i** **NOTE:** For all other types of ports that support 10G SFP+ connections (for example, SFP28 or ports with 10G SFP+ using a QSA adapter), do not enable autonegotiation or link training. There is no support for host-side autonegotiation and link training on 10G/1GBASE-T modules.

**i** **NOTE:** On N3248X-ON , N3248PXE-ON, and E3248PXE-ON switches, half-duplex is supported for 10 Mbps ,100 Mbps, and 1Gbps speeds.

To disable autonegotiation and restore the default port speed, enter `no speed auto`.

**Link training** — (Ethernet physical port interfaces) Configure an Ethernet port with high-speed links through a copper cable to optimally tune the hardware signals sent and received to a connected device. When enabled on both sides of a link, the link training handshake starts to dynamically tune the hardware signals. If autonegotiation is also enabled on both sides of the link, link training is performed only after the interface speed is negotiated. Link training avoids the need to manually tune specific channels in a copper cable link, including IEEE 802.3bj-2014. Link training is disabled by default, and is always enabled when autonegotiation is activated.

You can use standalone link training on an interface to enable link training when autonegotiation is disabled. Link training operational status on hardware transceivers is shown in the following table.

**Table 27. Link training operational status**

| Autonegotiation | Standalone Link Training | Operational Link Training Status |
|-----------------|--------------------------|----------------------------------|
| OFF             | OFF                      | OFF                              |
| OFF             | ON                       | ON (SFP, QSFP, and QSFP-DD only) |
| ON              | OFF                      | ON (SFP, QSFP, and QSFP-DD only) |

**Table 27. Link training operational status (continued)**

| <b>Autonegotiation</b> | <b>Standalone Link Training</b> | <b>Operational Link Training Status</b> |
|------------------------|---------------------------------|-----------------------------------------|
| ON                     | ON                              | ON (SFP, QSFP, and QSFP-DD only)        |

You can enable standalone link training on an interface when autonegotiation is off. In this case, link training remains on and autonegotiation is off. If you disable standalone link training, link training is disabled and autonegotiation remains off.

- `standalone-link-training` — Enables standalone link training on an interface.
- `no standalone-link-training` — Disables standalone link training on an interface.

**(i) NOTE:** Dell Technologies recommends that you configure link training on all Ethernet interfaces with copper cable connections. Without link-trained interfaces, an Enterprise SONiC switch requires statically tuned transmit (tx) signal parameters, which is difficult to scale and not supported on all switches.

**(i) NOTE:** Link training is only supported on links with speeds of 40G (40000 Mbps) or more. Disable link training on ports with native twisted-pair RJ45 copper cable connections.

**(i) NOTE:** Link training is not supported on the Z9264F-ON. Starting in Release 4.1.0, link training is supported on N3200-ON and E3200-ON series switches.

Link training requires a supported media type to be installed on a port. For information on the breakout modes, cables, and optics supported on each switch, contact your designated sales representative.

### Ethernet interface configuration and display

```
sonic(config)# interface Eth1/3
sonic(conf-if-Ethernet68)#
sonic(conf-if-Ethernet68)# description "Test Config"
sonic(conf-if-Ethernet68)# ip address 10.210.9.3/31
sonic(conf-if-Ethernet68)# ipv6 address 10:21:10:9::3/64
sonic(conf-if-Ethernet68)# ip access-group ipv4_ingress_acl in
sonic(conf-if-Ethernet68)# ipv6 access-group ipv6_ingress_acl in
sonic(conf-if-Ethernet68)# mtu 9100
sonic(conf-if-Ethernet68)# speed 10000
sonic(conf-if-Ethernet68)# no shutdown
sonic(conf-if-Ethernet68)#

sonic(config)# do show interface Eth1/3

Eth1/3 is up, line protocol is up, reason oper-up
Hardware is Eth, address is 0c:29:ef:e1:f8:02
Description: Test Config
IPV4 address is 10.210.9.3/31
Mode of IPV4 address assignment: MANUAL
IPV6 address is 10:21:10:9::3/64,fe80::e29:efff:fe1:f802/64
Mode of IPV6 address assignment: MANUAL
IP MTU 9100 bytes
LineSpeed 10GB, Auto-negotiation off
Link-training: off
Unreliable-LOS: off
FEC: DISABLED
Events:
 initialized at 2022-03-24T05:46:43.421001Z
 admin-up at 2022-03-24T05:46:53.361035Z
 xcvr-status-up at 2022-03-24T05:48:07.486835Z
 port-enabled at 2022-03-24T05:48:07.487593Z
 phy-link-up at 2022-03-24T05:48:07.729433Z
Last clearing of "show interface" counters: never
10 seconds input rate 70000 packets/sec, 769602688 bits/sec, 96200336 Bytes/sec
10 seconds output rate 19999 packets/sec, 242878752 bits/sec, 30359844 Bytes/sec
Input statistics:
 2606931547 packets, 3582259145792 octets
 20 Multicasts, 466699 Broadcasts, 2606464835 Unicasts
 0 error, 209297229 discarded, 0 Oversize
 6 Packets (128 to 255 Octects)
Output statistics:
 682419292 packets, 1035901416402 octets
 1877 Multicasts, 1 Broadcasts, 682417414 Unicasts
```

```

 0 error, 0 discarded, 0 Oversize
Time since last interface status change: 12:51:50

```

### Management interface configuration and display

**(i) NOTE:** If you use using the gwaddr option to configure the Management interface, all traffic originating from the Management IP address on the switch as well as all locally switched traffic on the Management interface is forwarded to the device with the configured gwaddr IP address. If this behavior is not desired, configure the Management VRF (ip vrf mgmt command) when you assign the gwaddr IP address to the Management interface. Dell Technologies recommends that you configure the Management VRF so that the Management interface is automatically added, and management traffic is fully isolated from other configured VRFs.

```

sonic(config)# interface Management 0
sonic(conf-if-Management0)# description Management0
sonic(conf-if-Management0)# ip address 100.94.189.13/24 gwaddr 100.94.189.254
sonic(conf-if-Management0)# ipv6 address 2000::13/64
sonic(conf-if-Management0)# mtu 1500
sonic(conf-if-Management0)# autoneg on
sonic(conf-if-Management0)# speed 1000
sonic(conf-if-Management0)# no shutdown
sonic(conf-if-Management0)#
sonic(conf-if-Management0)# do show interface Management 0
Management0 is up, line protocol is up
Hardware is MGMT, address is 0c:29:ef:e1:f8:00
Description: Management0
IPV4 address is 100.94.189.13/24
Mode of IPV4 address assignment: MANUAL
IPV6 address is 2000::13/64,fe80::e29:efff:feel:f800/64
Mode of IPV6 address assignment: DHCP
IP MTU 1500 bytes
LineSpeed 1GB, Auto-negotiation True
Input statistics:
 75106 packets, 15083074 octets
 28980 Multicasts, 0 error, 1553 discarded
Output statistics:
 3291 packets, 765626 octets
 0 error, 0 discarded
Time since last interface status change: 12:56:27

```

### View interface status and operation

```

sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/1 connected to spine up oper-up off 40000 1500 Ethernet0
Eth1/2 - down admin-down off 40000 9100 Ethernet4
Eth1/3 Ethernet L3 up oper-up off 40000 1500 Ethernet8
Eth1/4 Ethernet L2 up oper-up off 40000 9100 Ethernet12
...

```

**(i) NOTE:** To view the show output in one display without having to page through screen displays, enter show interface status | no-more.

```

sonic# show interface counters

Interface State RX_OK RX_ERR RX_DRP RX_OVERSIZE TX_OK TX_ERR TX_DRP TX_OVERSIZE

Eth1/1 D 0 0 0 0 0 0 0 0 0
Eth1/2 U 201809144 0 0 0 201809148 0 0 0 0
Eth1/3 D 0 0 0 0 0 0 0 0 0
Eth1/4 D 0 0 0 0 0 0 0 0 0
Eth1/5 D 0 0 0 0 0 0 0 0 0
Eth1/6 D 0 0 0 0 0 0 0 0 0
Eth1/7 D 0 0 0 0 0 0 0 0 0
Eth1/8 D 0 0 0 0 0 0 0 0 0
Eth1/9 U 201803931 0 0 0 201803929 0 0 0 0
Eth1/10 D 0 0 0 0 0 0 0 0 0

```

```
Eth1/11 D 0 0 0 0 0 0 0 0
```

```
...
sonic# clear counters interface Eth1/1
Clear counters for Eth1/1 [confirm y/N]: y
```

```
sonic# clear counters interface Eth1/2
Clear counters for Eth1/2 [confirm y/N]: y
```

### Configure and view L3 routing interface counters

The collection of L3 routing interface counters is enabled by default. To reconfigure the interval used to record IPv4/IPv6 traffic statistics — bytes per second (BPS) and packets per second (PPS) — on a routing interface (`rif`) in both the receive (RX) and transmit (TX) directions, use the `counters rif interval` command. Enter the polling interval in seconds (1 to 30; default 1). A routing interface includes the routing port, L3 interface and subinterfaces, and configured VLANs.

```
sonic(config)# counters rif interval seconds
```

```
sonic# show interface counters rif
Polling Rate : 5 seconds

Interface RX_OK RX_BPS RX_PPS RX_ERR TX_OK TX_BPS TX_PPS TX_ERR

Ethernet46 561 100 32 N/A 34678 364 941 N/A
Ethernet96 0 0 0 N/A 0 0 0 N/A
Po33.101 258 N/A 4 N/A 698 N/A N/A N/A
Vlan200 5432 58 123 N/A 0 0 0 N/A
PortChannel1256 0 0 0 N/A 4852 68 12 N/A
```

To clear all L3 interface counters:

```
sonic# clear counters interface rif
```

### Display loopback interface configuration

```
sonic# show interface Loopback 1
Loopback1 is up, line protocol is up
Hardware is Loopback, address is 0c:29:ef:e1:f8:02
IPV4 address is 50.1.1.1/32
Mode of IPV4 address assignment: MANUAL
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
Time since last interface status change: 13:05:17
```

### Configure link training

To enable link training on a port interface regardless of the autonegotiation setting — link training remains enabled if autonegotiation is on or off:

**i** **NOTE:** Link training is not enabled if per-port connector and cable settings are statically configured in the `media_settings.json` file.

**i** **NOTE:** Autonegotiation and link training are not supported on the 10G SFP+ ports of the Z9664F-ON, Z9432F-ON, Z9332F-ON, S5448F-ON, Z9264F-ON, and S5232F-ON switches. For all other types of ports that support 10G SFP+ connections (for example, SFP28), do not enable autonegotiation for 10G SFP+ links.

```
sonic(config)# interface Eth1/1
sonic(conf-if-eth1/1)# standalone-link-training

sonic(config)# interface range Eth1/1-1/15
%Info: Configuring only existing interfaces in range
sonic(conf-if-range-eth**)# standalone-link-training
```

To disable link training on a port interface, autonegotiation must be off. If autonegotiation is on, link training remains enabled after you enter the `no standalone-link-training` command.

**(i) NOTE:** Link training is disabled even if per-port connector and cable settings are statically configured in the media\_settings.json file.

```
sonic(config)# interface Eth1/1
sonic(conf-if-eth1/1)# no standalone-link-training

sonic(config)# interface range Eth1/1-1/15
%Info: Configuring only existing interfaces in range
sonic(conf-if-range-eth**)# no standalone-link-training
```

## Configure speed

When you set the speed that is advertised by a port interface, you disable auto-negotiation on the port. For example, to set a transmission speed of 100G and disable autonegotiation:

```
sonic(conf-if-Eth1/1)# speed 100000
```

To enable autonegotiated speed, enter `speed auto`. To advertise a specific speed to autonegotiate with connecting devices:

```
sonic(conf-if-Eth1/1)# speed auto 100000
```

**(i) NOTE:** Autonegotiation is not supported on the Z9264F-ON.

To restore the default port speed:

```
sonic(conf-if-Eth1/1)# no speed
```

To disable autonegotiation and restore the default port speed:

```
sonic(conf-if-Eth1/1)# no speed auto
```

To disable autonegotiation and set a specified port speed, such as 100G:

```
sonic(conf-if-Eth1/1)# no speed auto 100000
```

## View link and autonegotiation status

```
sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/1 - up oper-up off 100000 9100 Ethernet0
Eth1/2 - down admin-down off 100000 9100 Ethernet1
Eth1/3 - down admin-down off 100000 9100 Ethernet2
Eth1/4 - up oper-up off 100000 9100 Ethernet3
Eth1/5 - down admin-down off 100000 9100 Ethernet4
Eth1/6 - down admin-down off 100000 9100 Ethernet5
...
...
```

## View link training status

```
sonic# show interface link-training

Interface Type Auto Negotiation Link-Training
----- -----
Eth1/1/1 QSFP56-DD off off off
Eth1/2 QSFP56-DD off on not_trained
Eth1/3/1 QSFP28 on off trained
Eth1/4/1 QSFP56-DD off off off
Eth1/4/2 QSFP56-DD off off off
Eth1/4/3 QSFP56-DD off off off
Eth1/4/4 QSFP56-DD off off off
Eth1/5 QSFP56-DD off off off
Eth1/6/1 QSFP56-DD on off trained
Eth1/6/2 QSFP56-DD on off trained
Eth1/6/3 QSFP56-DD on off trained
Eth1/6/4 QSFP56-DD on off trained
Eth1/7 QSFP56-DD off off off
```

```

Eth1/8/1 QSFP28-DD 2x100GBASE-2SR4-DUALRATE off off off
Eth1/8/2 QSFP28-DD 2x100GBASE-2SR4-DUALRATE off off off
Eth1/8/3 QSFP28-DD 2x100GBASE-2SR4-DUALRATE off off off
Eth1/8/4 QSFP28-DD 2x100GBASE-2SR4-DUALRATE off off off
Eth1/8/5 QSFP28-DD 2x100GBASE-2SR4-DUALRATE off off off
Eth1/8/6 QSFP28-DD 2x100GBASE-2SR4-DUALRATE off off off
Eth1/8/7 QSFP28-DD 2x100GBASE-2SR4-DUALRATE off off off
Eth1/8/8 QSFP28-DD 2x100GBASE-2SR4-DUALRATE off off off
Eth1/9/1 QSFP28 100GBASE-SR4-AOC-10.0M off off off
...

```

```

sonic# show interface link-training Eth 1/3/1
 Auto Link-Training
Interface Type Negotiation Standalone Operational

Eth1/3/1 QSFP28 100GBASE-CR4-DAC-1.0M on off trained

```

```

sonic# show interface link-training Eth 1/2,1/3/1,1/31
 Auto Link-Training
Interface Type Negotiation Standalone Operational

Eth1/2 QSFP56-DD 400GBASE-CR8-DAC-2.0M off on not_trained
Eth1/3/1 QSFP28 100GBASE-CR4-DAC-1.0M on off trained
Eth1/31
 off off off

```

### **View advertised port speeds and autonegotiated settings**

Use the `show interface advertise` command to display the configured local administrative link advertisement, local operational link advertisement, and the link-partner advertisement for a specified interface. If a link is down, the `Oper Peer Advertisement` displays a dash.

```
sonic# show interface advertise [Ethsslot/port]
```

```
sonic# show interface advertise Eth1/3/1
```

```

Name: Eth1/3/1
Type: QSFP28 100GBASE-CR4-DAC-1.0M
Admin State: UP
Link Status: UP
Auto Negotiation: ON
Operational FEC Mode: RS
Operational Link Training: TRAINED
Standalone Link Training: OFF

 400G 200G 100G 50G 40G 25G 10G 5G 2.5g 1G 100f 100h 10f 10h
 ----- -----
Admin Local no no yes no yes no no no no no no no no no
Advertisement
Oper Local no no yes no no
Advertisement
Oper Remote no no yes no no
Advertisement

```

If you use the command without a specified interface, the autonegotiation state and operational local link advertisement are displayed for all ports. The operational advertised speed displays a value only if the speed is supported on both the local switch and the link-partner device. If autonegotiation is disabled, the operational advertised speed is not displayed.

```

sonic# show interface advertise

Name Type Auto-Neg Oper Advertised Speed
----- -----
Eth1/1 QSFP28 100GBASE-SR4 on 40000,100000
Eth1/2
 on 40000,100000
Eth1/3 QSFP28 100GBASE-SR4 on 40000,100000
Eth1/4
 on 40000,100000
...

```

# Configure access and trunk interfaces

By default, an Ethernet or port channel interface is not in L2 mode. Configure the interface to operate either as a trunk (IEEE 802.1Q encapsulated traffic) or access (untagged traffic) port:

## Access ports

An access port sends and receives untagged frames from connected devices. By default, no access VLAN is assigned to a port interface.

- To assign an interface to an access VLAN, use the `switchport access Vlan vlan-id` command.

```
switchport access Vlan vlan-id
```

- To remove an interface from the access VLAN, enter the `no switchport access Vlan` command.

## Add interface as access port to a VLAN

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# switchport access Vlan 100
sonic(conf-if-Eth1/1)# end

sonic# show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Active A Eth1/1 Enable No

sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# no switchport access Vlan

sonic(conf-if-Eth1/1)# do show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Inactive None Enable No
```

## Trunk ports

A trunk port sends and receives tagged frames from multiple VLANs and untagged frames from the access VLAN. By default, no trunk VLANs are assigned to a port interface. You can remove VLAN IDs from the allowed VLANs, and later add VLAN IDs to the list to allow their traffic to pass on the trunk.

To configure the allowed VLAN traffic on a trunk interface, use the `switchport trunk allowed Vlan` command.

```
switchport trunk allowed Vlan {vlan-list | {add | remove | except} vlan-list | none | all}
```

- *vlan-list* — Specifies the VLAN IDs to be added or removed from the list of allowed VLAN traffic. Enter a VLAN range with a hyphen (-); for example, 10-21. Separate VLAN IDs and VLAN ranges using a comma (,); for example, 10,15-18,21.
- *add* — Adds the specified VLAN IDs to the list of allowed VLAN traffic.
- *remove* — Removes the specified VLAN IDs from the list of allowed VLAN traffic and prevents the removed VLAN traffic from being carried on the interface. To remove a tagged VLAN membership, enter the `switchport trunk allowed Vlan remove vlan-list` command.
- *except* — Adds all VLANs to the list of allowed VLAN traffic, except for the specified VLAN IDs.
- *none* — Removes all trunk VLANs from the allowed VLAN list.
- *all* — Adds all VLAN IDs to the allowed VLAN list.

To remove all VLANs (tagged and untagged) from an interface, enter the `no switchport allowed Vlan` command.

## Add interface as trunk port to a VLAN

```
sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# switchport trunk allowed Vlan add 100
sonic(conf-if-Eth1/3)# end

sonic# show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
```

|     |        |          |        |    |
|-----|--------|----------|--------|----|
| 100 | Active | T Eth1/3 | Enable | No |
|     |        | A Eth1/1 |        |    |

(i) **NOTE:** In show Vlan output:

- NUM — VLAN ID number
- Status — VLAN status displays as Active or Inactive.
  - Active — A VLAN member is present and the line protocol for at least one VLAN member is up.
  - Inactive — No VLAN member is present or the line protocol for all VLAN members is down.
- Q — Displays the 802.1Q mode of a VLAN member interface:
  - T — Tagged VLAN member
  - A — Access VLAN member
- Autostate — Displays the VLAN autostate mode: Enable or Disable.
- Dynamic — Yes indicates a RADIUS-supplied VLAN (see **RADIUS-supplied VLANs** in [Port access control](#)); No indicates a static VLAN that was manually configured.

#### Add interface as trunk port to all VLANs

```
sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# switchport trunk allowed vlan all

sonic# show running-configuration interface Eth1/3
!
interface Eth1/3
 mtu 9100
 speed 40000
 shutdown
 switchport trunk allowed Vlan 1-4094
```

#### Add interface as trunk port to all VLANs except specified VLANs

```
sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# switchport trunk allowed vlan except 1000-4000

sonic# show running-configuration interface Eth1/3
!
interface Eth1/3
 mtu 9100
 speed 40000
 shutdown
 switchport trunk allowed Vlan 1-999,4001-4094
```

#### Add interface as trunk port to no VLANs

```
sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# switchport trunk allowed vlan none

sonic# show running-configuration interface Eth1/3
!
interface Eth1/3
 mtu 9100
 speed 40000
 shutdown
```

#### Add interface as trunk port to VLAN range

```
! Create VLAN range
sonic(config)# interface range create Vlan 100-103,110
sonic(conf-if-range-vl**)# end

sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# switchport trunk allowed vlan add 101-103,110
sonic(conf-if-Eth1/1)# end

sonic# show running-configuration interface Eth1/1
!
```

```

interface Eth1/1
no shutdown
switchport access Vlan 100
switchport trunk allowed Vlan 101-103,110

sonic# show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Active A Eth1/1 Enable No
101 Active T Eth1/1 Enable No
102 Active T Eth1/1 Enable No
103 Active T Eth1/1 Enable No
104 Inactive - Enable No
105 Inactive - Enable No
110 Active T Eth1/1 Enable No

```

#### Add port channel as trunk port to a VLAN

```

sonic(config)# interface PortChannel 12
sonic(conf-if-po12)# switchport trunk allowed Vlan add 100
sonic(conf-if-po12)# end

sonic# show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Active T Eth1/3 Enable No
 T PortChannel12
 A Eth1/1
101 Active T Eth1/1 Enable No
102 Active T Eth1/1 Enable No
103 Active T Eth1/1 Enable No
104 Inactive - Enable No
105 Inactive - Enable No
110 Active T Eth1/1 Enable No

```

#### Remove interface as trunk port from VLAN range

```

sonic(conf-if-Eth1/1)# show configuration
!
interface Eth1/1
no shutdown
switchport access Vlan 100
switchport trunk allowed Vlan 101-103,110

sonic(conf-if-Eth1/1)# no switchport trunk allowed Vlan 101-103

sonic(conf-if-Eth1/1)# show configuration
!
interface Eth1/1
no shutdown
switchport access Vlan 100
switchport trunk allowed Vlan 110

```

#### Add/Remove port channel as trunk port to/from VLAN range

```

sonic(config)# do show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Active T Eth1/3 Enable No
 T PortChannel12
 A Eth1/1
101 Inactive - Enable No
102 Inactive - Enable No
103 Inactive - Enable No
104 Inactive - Enable No
105 Inactive - Enable No
110 Active T Eth1/1 Enable No

sonic(config)# interface PortChannel 12
sonic(conf-if-po12)# switchport trunk allowed Vlan add 100-105

```

```

sonic(conf-if-po12)# show configuration
!
interface PortChannel 12 mode on
switchport trunk allowed Vlan 100-105
no shutdown

sonic(conf-if-po12)# do show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Active T Eth1/3 Enable No
 T PortChannel12
 A Eth1/1
101 Active T PortChannel12 Enable No
102 Active T PortChannel12 Enable No
103 Active T PortChannel12 Enable No
104 Active T PortChannel12 Enable No
105 Active T PortChannel12 Enable No
110 Active T Eth1/1 Enable No

sonic(conf-if-po12)# switchport trunk allowed Vlan remove 100-105
sonic(conf-if-po12)# do show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Active T Eth1/3 Enable No
 A Eth1/1
101 Inactive T
102 Inactive T
103 Inactive T
104 Inactive T
105 Inactive T
110 Active T Eth1/1 Enable No

```

### Remove all VLAN configurations

```

sonic# show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Active T Eth1/3 Enable No
 T PortChannel12
 A Eth1/1
101 Active T PortChannel12 Enable No
102 Active T PortChannel12 Enable No
103 Active T PortChannel12 Enable No
104 Active T PortChannel12 Enable No
105 Active T PortChannel12 Enable No
110 Active T Eth1/1 Enable No
 A PortChannel12

sonic# configure terminal
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# no switchport allowed Vlan
sonic(conf-if-Eth1/1)# exit

sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# no switchport allowed Vlan
sonic(conf-if-Eth1/3)# exit

sonic(config)# interface PortChannel 12
sonic(conf-if-po12)# no switchport allowed Vlan

sonic(conf-if-po12)# do show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Inactive T
101 Inactive T
102 Inactive T
103 Inactive T
104 Inactive T
105 Inactive T
110 Inactive T

```

# Interface ranges

To view or configure multiple interfaces using one command, enter an interface range in the configuration or show command syntax. Use an interface range to specify multiple Ethernet, VLAN, or port channel interfaces.

- Use the `interface range` command to configure a range of existing interfaces.
- Use the `interface range create` command to create any nonexisting VLAN or port channel interfaces in the specified range, and enter Interface range configuration mode for all specified interfaces.
- You can enter multiple interface ranges in a configuration or show command. Separate each range or individual interface number with a comma.
- Interface ranges are supported in both native and standard interface-naming modes.
- Various configuration commands are supported in interface-range configuration mode:
  - [no] `channel-group`
  - `no ip address`
  - [no] `ip vrf`
  - [no] `ipv6 enable`
  - [no] `mtu`
  - [no] `shutdown`
  - [no] `speed`
  - [no] `switchport`
- Interface ranges are not supported in gNMI and REST API operations.

## Configure an Ethernet interface range

```
sonic(config)# interface range Eth slot/port[/breakout-port]-slot/port[/breakout-port]
[,slot/port[/breakout-port]-slot/port[/breakout-port] ...]
sonic(conf-if-range-eth**)#
```

For example:

```
sonic(config)# interface range Eth 1/1-1/2,1/7,1/12-1/15
%Info: Configuring only existing interfaces in range
sonic(conf-if-range-eth**)# mtu 9000
```

These Ethernet interface range syntaxes are supported in the `interface range` command:

```
sonic(config)# interface range Eth1/1-1/2,1/7,1/12-1/15
%Info: Configuring only existing interfaces in range
sonic(conf-if-range-eth**)# mtu 9000
```

## View an Ethernet interface range

 **NOTE:** Interface information displays only for configured interfaces in the specified range.

```
sonic# show interface Eth1/1-1/2,1/4,1/7-1/9

Eth1/1 is down, line protocol is down
Hardware is Eth
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 400GB, Auto-negotiation off
FEC: RS
Last clearing of "show interface" counters: never
10 seconds input rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
10 seconds output rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
Input statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
 0 Packets (128 to 255 Octects)
Output statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
```

```

 0 error, 0 discarded, 0 Oversize
Eth1/2 is down, line protocol is down
Hardware is Eth
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 400GB, Auto-negotiation off
FEC: RS
Last clearing of "show interface" counters: never
10 seconds input rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
10 seconds output rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
Input statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
 0 Packets (128 to 255 Octects)
Output statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
Eth1/4 is down, line protocol is down
Hardware is Eth
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 400GB, Auto-negotiation off
FEC: RS
Last clearing of "show interface" counters: never
10 seconds input rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
10 seconds output rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
Input statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
 0 Packets (128 to 255 Octects)
Output statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
Eth1/7 is down, line protocol is down
Hardware is Eth
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 400GB, Auto-negotiation off
FEC: RS
Last clearing of "show interface" counters: never
10 seconds input rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
10 seconds output rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
Input statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
 0 Packets (128 to 255 Octects)
Output statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
Eth1/8 is down, line protocol is down
Hardware is Eth
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 400GB, Auto-negotiation off
FEC: RS
Last clearing of "show interface" counters: never
10 seconds input rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
10 seconds output rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
Input statistics:

```

```

 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
 0 Packets (128 to 255 Octects)
Output statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
Eth1/9 is down, line protocol is down
Hardware is Eth
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 400GB, Auto-negotiation off
FEC: RS
Last clearing of "show interface" counters: never
10 seconds input rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
10 seconds output rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
Input statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
 0 Packets (128 to 255 Octects)
Output statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize

```

### Configure a VLAN range

```

sonic(config)# interface range Vlan vlan-id - vlan-id[,vlan-id - vlan-id ...]
sonic(conf-if-range-vl**)#

```

For example:

```

sonic(config)# interface range Vlan 1-10,12,20-21
sonic(conf-if-range-vl**)# ipv6 enable

```

### Create and configure VLAN range

To create any nonexisting VLAN interfaces in a specified range and configure all VLAN interfaces in the range:

```

sonic(config)# interface range create Vlan vlan-id - vlan-id[,vlan-id - vlan-id ...]
sonic(conf-if-range-vl**)#

```

For example:

```

sonic(config)# interface range create Vlan 1-10,12,20-21
sonic(conf-if-range-vl**)# ipv6 enable

```

### View a VLAN range

```

sonic# show interface Vlan 1-10,12

Vlan1 is up
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Enabled
IP MTU 9100 bytes

Vlan2 is up
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Enabled
IP MTU 9100 bytes
...
Vlan12 is up
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set

```

```
Interface IPv6 oper status: Enabled
IP MTU 9100 bytes
```

### Delete a VLAN range

```
sonic(config)# no interface range Vlan 1-12
sonic(config)#
```

### Configure a port-channel range

```
sonic(config)# interface range PortChannel portchannel-number - portchannel-number[,portchannel-number - portchannel-number ...]
sonic(conf-if-range-po**)#
```

For example:

```
sonic(config)# interface range PortChannel 1-2,7-8,10
sonic(conf-if-range-po**)# mtu 6000
```

### Create and configure port-channel range

To create any nonexisting port channel interfaces in a specified range and configure all port channel interfaces in the range:

```
sonic(config)# interface range create PortChannel portchannel-number - portchannel-number[,portchannel-number - portchannel-number ...]
sonic(conf-if-range-po**)#
```

For example:

```
sonic(config)# interface range create PortChannel 3-4,10,12
sonic(conf-if-range-po**)#
```

### View a port channel range

```
sonic# show interface PortChannel 3-4,12

PortChannel3 is up, line protocol is down, mode LACP
Minimum number of links to bring PortChannel up is 1
Fallback: Disabled
Graceful shutdown: Disabled
MTU 9100
LACP mode ACTIVE interval SLOW priority 65535 address 20:04:0f:1d:22:4e

PortChannel4 is up, line protocol is down, mode LACP
Minimum number of links to bring PortChannel up is 1
Fallback: Disabled
Graceful shutdown: Disabled
MTU 9100
LACP mode ACTIVE interval SLOW priority 65535 address 20:04:0f:1d:22:4e

PortChannel12 is up, line protocol is down, mode LACP
Minimum number of links to bring PortChannel up is 1
Fallback: Disabled
Graceful shutdown: Disabled
MTU 9100
LACP mode ACTIVE interval SLOW priority 65535 address 20:04:0f:1d:22:4e
```

### Delete a port channel range

```
sonic(config)# no interface range Port-Channel 1-2,10
```

## Forward error correction

Forward error correction (FEC) allows a switch to send redundant parity packets that a receiver uses to reconstruct lost data. If there is a transmission error, FEC avoids the need to retransmit the data. Enable FEC on both ends of a link. Enterprise SONiC supports FEC types FC and RS.

By default, FEC is not enabled on an interface, except for 400G, 4x100G, 1x200G and 2x200G interfaces on which FEC RS is enabled. To configure FEC on 25G and 100G interfaces, use the `fec` command.

A FEC configuration mismatch on both sides of a link results in a link-down state. To reduce these errors, configure automatic FEC configuration based on the installed transceiver that is detected on a port.

**i | NOTE:** When enabled, port autonegotiation (also known as IEEE Autonegotiation+LinkTraining) automatically enables a FEC type for DAC copper interfaces. To ensure automatic FEC configuration, enable auto-FEC. When enabled, auto-FEC enables a FEC type for DAC copper interfaces. If both auto-FEC and autonegotiation are enabled on a port interface, autonegotiation takes precedence.

**i | NOTE:** Autonegotiation also takes precedence if configured on an interface with `fec rs` or `fec fc` already configured.

## Configure FEC

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# fec {auto | FC | RS | none}
```

- `auto` — Automatically detects an installed transceiver and configures the correct FEC type on a port.
- `FC` — Enables FEC type FC on supported interfaces. FC stands for fire code.
- `RS` — Enables FEC type RS on supported interfaces. RS stands for Reed-Solomon code.
- `none` — Disables FEC on an interface.

Enter the `no fec` command to reset FEC to its system default value.

## View FEC configuration (25G port)

```
sonic# show interface Eth 1/71

Eth1/71 is up, line protocol is up, reason oper-up
Hardware is Eth, address is 3c:2c:30:29:78:82
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 25GB, Auto-negotiation off
Link-training: off
Unreliable-LOS: off
FEC: FC
Events:
 initialized at 2023-03-03T23:02:23.378951Z
 admin-up at 2023-03-03T23:02:28.338271Z
 xcvr-status-up at 2023-03-03T23:03:26.12158Z
 port-enabled at 2023-03-03T23:03:26.122019Z
 pcs-errors at 2023-03-07T10:50:23.228555Z
 remote-fault at 2023-03-07T13:47:30.288484Z
 phy-link-down at 2023-03-07T14:23:54.009714Z
 local-fault at 2023-03-07T14:23:54.017398Z
 phy-link-up at 2023-03-07T14:25:21.570409Z
Last clearing of "show interface" counters: 2023-03-07 21:16:12+0000
10 seconds input rate 670305 packets/sec, 24878039240 bits/sec, 3109754905 Bytes/sec
10 seconds output rate 670390 packets/sec, 24873913944 bits/sec, 3109239243 Bytes/sec
Input statistics:
 3358699 packets, 15588130307 octets
 0 Multicasts, 0 Broadcasts, 3358696 Unicasts
 0 error, 0 discarded, 0 Oversize
 41656 Packets (128 to 255 Octects)
Output statistics:
 3358153 packets, 15583054984 octets
 0 Multicasts, 0 Broadcasts, 3358153 Unicasts
 0 error, 0 discarded, 0 Oversize
Time since last interface status change: 06:50:56
```

```
sonic# show running-configuration interface Eth 1/71
!
interface Eth1/71
 mtu 9100
 speed 25000
 fec FC
```

```
unreliable-los auto
no shutdown
```

### View FEC configuration (100G port)

```
sonic# show interface Eth 1/101

Eth1/101 is up, line protocol is up, reason oper-up
Hardware is Eth, address is 3c:2c:30:29:78:82
Description: to 100G Ixia
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9216 bytes
LineSpeed 100GB, Auto-negotiation on
Link-training: trained
Unreliable-LOS: off
FEC: RS
Events:
 initialized at 2023-03-03T23:02:23.759296Z
 admin-up at 2023-03-03T23:02:27.322959Z
 xcvr-status-up at 2023-03-03T23:03:26.051153Z
 port-enabled at 2023-03-03T23:03:26.051856Z
 phy-link-up at 2023-03-03T23:03:28.368728Z
Last clearing of "show interface" counters: 2023-03-07 21:16:12+0000
10 seconds input rate 1067607 packets/sec, 39639440096 bits/sec, 4954930012 Bytes/sec
10 seconds output rate 1067604 packets/sec, 39630420392 bits/sec, 4953802549 Bytes/sec
Input statistics:
 268140297 packets, 1244161628567 octets
 0 Multicasts, 0 Broadcasts, 268140297 Unicasts
 0 error, 0 discarded, 0 Oversize
 3742238 Packets (128 to 255 Octects)
Output statistics:
 268140317 packets, 1244213994798 octets
 8 Multicasts, 0 Broadcasts, 268140309 Unicasts
 0 error, 0 discarded, 0 Oversize
Time since last interface status change: 3d22h16m
```

```
sonic# show running-configuration interface Eth 1/101
!
interface Eth1/101
 mtu 9216
 speed auto
fec RS
 unreliable-los auto
 no shutdown
```

### View FEC operational status on a port range

```
sonic# show interfaces fec status [Ethslot/port[/breakout-port]] [port-range]
```

```
show interface fec status Eth 1/4,1/23-1/26,1/28
Interface Type Oper Admin If-State
----- -----
Eth1/4 SFP+ 10GBASE-CR-DAC-7.0M none none oper-up
Eth1/23 SFP28 25GBASE-CR-DAC-1.0M rs auto oper-up
Eth1/24 SFP28 25GBASE-CR-DAC-1.0M rs auto oper-up
Eth1/25 QSFP28 100GBASE-CR4-DAC-5.0M rs rs oper-up
Eth1/26 QSFP+ 40GBASE-CR4-DAC-7.0M none none admin-down
Eth1/28 QSFP28 100GBASE-SR4-AOC-30.0M rs auto oper-up
```

### Configure FEC on a transceiver

For certain media types, such as the QSFP56-DD 400BASE-SR4.2, FEC runs on the installed transceiver instead of on the port interface. After you break out a 4x100G port — see [Port breakouts](#) — the port's configuration, including the FEC setting, is deleted. You must configure FEC on the port; for example:

```
sonic(config)# interface breakout port Eth1/6 mode 4x100G
sonic(config)# interface media-fec port 1/6 mode {ieee | custom}
```

- Enter `ieee` to enable the IEEE KP/KP4 RS FEC mode on a supported transceiver (default).
- Enter `custom` to enable a proprietary or custom FEC mode on a supported transceiver; for example, the BRCM proprietary KP/KP4 RS FEC mode on the Q56DD-400G-SR4.2 transceiver.

To verify the FEC mode on breakout interfaces, enter the `show interface` command; for example:

```
Z9432F-04# show interface Eth 1/14/1

Eth1/14/1 is up, line protocol is up
Hardware is Eth
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 100GB, Auto-negotiation off
Link-training: off
Unreliable-LOS: off
Media FEC: CUSTOM
Last clearing of "show interface" counters: 2021-07-12 21:56:17
10 seconds input rate 2.590244e+06 packets/sec, 9.9586113928e+10 bits/sec,
1.2448264241e+10 Bytes/sec
10 seconds output rate 2.590239e+06 packets/sec, 9.9586094408e+10 bits/sec,
1.2448261801e+10 Bytes/sec
Input statistics:
 196120610706 packets, 942556775268464 octets
 0 Multicasts, 0 Broadcasts, 196120610706 Unicasts
 0 error, 0 discarded, 0 Oversize
 0 Packets (128 to 255 Octects)
Output statistics:
 196120613155 packets, 942556782693514 octets
 0 Multicasts, 0 Broadcasts, 196120613155 Unicasts
 0 error, 0 discarded, 0 Oversize
```

## Port groups

Port groups allow you to apply common configurations on a set of ports according to their hardware characteristics. Some Dell PowerSwitch switches have groups of ports that must share the same 10G or 25G speed. For example, in port group 2 that runs at 25G, to change one of the Ethernet interfaces from 25G to 10G, you must change all interfaces in the port group to 10G using the `port-group 2 speed 10000` command.

**i | NOTE:** Only the S5248F-ON and S5296F-ON switches support port groups.

### Display port groups

Use the `show port-group` command to display the port groups and member interfaces on a switch. The configurable interface speeds also displayed:

```
sonic# show port-group

Port-group Interface range Valid speeds Default Speed Current

1 Eth1/1 - Eth1/4 10G, 25G 25G 10G
2 Eth1/5 - Eth1/8 10G, 25G 25G 10G
3 Eth1/9 - Eth1/12 10G, 25G 25G 25G
4 Eth1/13 - Eth1/16 10G, 25G 25G 25G
5 Eth1/17 - Eth1/20 10G, 25G 25G 25G
6 Eth1/21 - Eth1/24 10G, 25G 25G 25G
7 Eth1/25 - Eth1/28 10G, 25G 25G 10G
8 Eth1/29 - Eth1/32 10G, 25G 25G 10G
9 Eth1/33 - Eth1/36 10G, 25G 25G 10G
10 Eth1/37 - Eth1/40 10G, 25G 25G 10G
11 Eth1/41 - Eth1/44 10G, 25G 25G 25G
12 Eth1/45 - Eth1/48 10G, 25G 25G 25G
```

### Configure port group speed

Use the `port-group speed` command to change the speed of the interfaces in a port group. The port speeds are in Megabits per second (Mbps).

```
sonic(config)# port-group number speed {10000 | 25000}

sonic(config)# port-group 1 speed 10000

sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/1 - up oper-up off 10000 9100 Ethernet0
Eth1/2 - up oper-up off 10000 9100 Ethernet1
Eth1/3 - up oper-up off 10000 9100 Ethernet2
Eth1/4 - up oper-up off 10000 9100 Ethernet3
...
...
```

**(i) NOTE:** Although the `port-group speed` command offers various port speed options, the only port group speeds that are supported are 10000 and 25000. If you configure an unsupported port speed, an error message displays:

```
sonic(config)# port-group 2 speed 1000
%Error: Unsupported speed
```

Port group interfaces that run at 10G can be individually configured to run at 1G. To configure a port to run at 1G, configure its port group speed as 10000. Then configure any individual interface in the port group to run at 1G using the `speed 1000` command. 1000 Mbps equals 1 Gbps.

```
sonic(config)# port-group 2 speed 10000
sonic(config)# interface Eth1/5
sonic(conf-if-Eth1/5)# speed 1000
sonic(conf-if-Eth1/5)# exit
sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/5 - up oper-up off 1000 9100 Ethernet4
...
...
```

**(i) NOTE:** If a port group is configured to run at 25G , you cannot configure an interface to run at 1G/1000 Mbps. You must first configure its port group speed as 10000:

```
sonic(config)# port-group 3 speed 25000
sonic(config)# interface Eth1/9
sonic(conf-if-Eth1/9)# speed 1000
%Error: Port group member. Please use port group command to change the speed

sonic(conf-if-Eth1/9)# port-group 3 speed 10000
sonic(config)# interface Eth1/9
sonic(conf-if-Eth1/9)# speed 1000
sonic(conf-if-Eth1/9)# exit
sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/9 - up oper-up off 1000 9100 Ethernet8
...
...
```

### Unconfigure port group speed

To unconfigure a port group's speed and set the default group speed, enter the `no port-group speed` command.

```
sonic(config)# no port-group 1 speed
sonic(config)# do show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/1 - up oper-up off 25000 9100 Ethernet0
Eth1/2 - up oper-up off 25000 9100 Ethernet1
...
...
```

|        |   |    |         |     |       |      |           |
|--------|---|----|---------|-----|-------|------|-----------|
| Eth1/3 | - | up | oper-up | off | 25000 | 9100 | Ethernet2 |
| Eth1/4 | - | up | oper-up | off | 25000 | 9100 | Ethernet3 |

### Supported port groups

To display the supported port groups on a switch, use the `show port-group` command. Only the S5248F-ON and S5296F-ON switches support port groups.

- S5248F-ON port groups
- S5296F-ON port groups

## S5296F-ON port groups

**Table 28. S5296F-ON port groups**

| Dell PowerSwitch | Port groups supported                                                                                                                                                                                                                                                                                                                                                           | Member interfaces                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Port-group speeds supported                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Default speed                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| S5296F-ON        | <ul style="list-style-type: none"> <li>• 1</li> <li>• 2</li> <li>• 3</li> <li>• 4</li> <li>• 5</li> <li>• 6</li> <li>• 7</li> <li>• 8</li> <li>• 9</li> <li>• 10</li> <li>• 11</li> <li>• 12</li> <li>• 13</li> <li>• 14</li> <li>• 15</li> <li>• 16</li> <li>• 17</li> <li>• 18</li> <li>• 19</li> <li>• 20</li> <li>• 21</li> <li>• 22</li> <li>• 23</li> <li>• 24</li> </ul> | <ul style="list-style-type: none"> <li>• Eth1/1 - Eth1/4</li> <li>• Eth1/5 - Eth1/8</li> <li>• Eth1/9 - Eth1/12</li> <li>• Eth1/13 - Eth1/16</li> <li>• Eth1/17 - Eth1/20</li> <li>• Eth1/21 - Eth1/24</li> <li>• Eth1/25 - Eth1/28</li> <li>• Eth1/29 - Eth1/32</li> <li>• Eth1/33 - Eth1/36</li> <li>• Eth1/37 - Eth1/40</li> <li>• Eth1/41 - Eth1/44</li> <li>• Eth1/45 - Eth1/48</li> <li>• Eth1/49 - Eth1/52</li> <li>• Eth1/53 - Eth1/56</li> <li>• Eth1/57 - Eth1/60</li> <li>• Eth1/61 - Eth1/64</li> <li>• Eth1/65 - Eth1/68</li> <li>• Eth1/69 - Eth1/72</li> <li>• Eth1/73 - Eth1/76</li> <li>• Eth1/77 - Eth1/80</li> <li>• Eth1/81 - Eth1/84</li> <li>• Eth1/85 - Eth1/88</li> <li>• Eth1/89 - Eth1/92</li> <li>• Eth1/93 - Eth1/96</li> </ul> | <ul style="list-style-type: none"> <li>• 10G, 25G</li> </ul> | <ul style="list-style-type: none"> <li>• 25G</li> </ul> |

## S5248F-ON port groups

**Table 29. S5248F-ON port groups**

| Dell PowerSwitch | Port groups supported                                                                                                                                         | Member interfaces                                                                                                                                                                                                                                                                                        | Port-group speeds supported                                                                                                                                                                              | Default speed                                                                                                                                                    |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| S5248F-ON        | <ul style="list-style-type: none"> <li>• 1</li> <li>• 2</li> <li>• 3</li> <li>• 4</li> <li>• 5</li> <li>• 6</li> <li>• 7</li> <li>• 8</li> <li>• 9</li> </ul> | <ul style="list-style-type: none"> <li>• Eth1/1 - Eth1/4</li> <li>• Eth1/5 - Eth1/8</li> <li>• Eth1/9 - Eth1/12</li> <li>• Eth1/13 - Eth1/16</li> <li>• Eth1/17 - Eth1/20</li> <li>• Eth1/21 - Eth1/24</li> <li>• Eth1/25 - Eth1/28</li> <li>• Eth1/29 - Eth1/32</li> <li>• Eth1/33 - Eth1/36</li> </ul> | <ul style="list-style-type: none"> <li>• 10G, 25G</li> </ul> | <ul style="list-style-type: none"> <li>• 25G</li> </ul> |

**Table 29. S5248F-ON port groups**

| Dell PowerSwitch | Port groups supported                                                              | Member interfaces                                                                                                               | Port-group speeds supported                                                                          | Default speed                                                                         |
|------------------|------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|                  | <ul style="list-style-type: none"> <li>• 10</li> <li>• 11</li> <li>• 12</li> </ul> | <ul style="list-style-type: none"> <li>• Eth1/37 - Eth1/40</li> <li>• Eth1/41 - Eth1/44</li> <li>• Eth1/45 - Eth1/48</li> </ul> | <ul style="list-style-type: none"> <li>• 10G, 25G</li> <li>• 10G, 25G</li> <li>• 10G, 25G</li> </ul> | <ul style="list-style-type: none"> <li>• 25G</li> <li>• 25G</li> <li>• 25G</li> </ul> |

## Port profiles

A port profile determines the enabled front-panel ports and supported breakout modes on Ethernet ports. The Z9432F-ON supports the default port profile and a nondefault profile. All other Dell PowerSwitch platforms that run Enterprise SONiC support only the default port profile; nondefault port profiles are not supported.

To view the breakout modes supported by the current port profile, use the `show interface breakout modes` command — see [Port breakouts](#).

### Using Z9432F-ON port profiles

- Display the port profiles that are supported on a Z9432F-ON by logging in to the switch. From the Linux shell, enter the command:

```
admin@z9432f:~$# sudo config-hwsku list
List of supported HWSKU:
DellEMC-Z9432f-O32 (Present)
DellEMC-Z9432f-T64C64O8-DPB
```

The default profile is installed (Present).

- To configure the nondefault port profile, enter the following command, which reboots the switch:

```
admin@z9432f:~$# sudo config-hwsku set DellEMC-Z9432f-T64C64O8-DPB
Warning: This deletes existing configurations and cause switch to reboot.
Are you sure you want to change? [y/N] y

Setting Default HWSKU to DellEMC-Z9432f-T64C64O8-DPB t1
Removing existing config_db.json
Rebooting system
Config Down success!
supervisor-proc-exit-listener: stopped
```

- After the switch reboots, log in and verify that the new port profile is installed.

```
admin@z9432f:~# sudo config-hwsku list
List of supported HWSKU:
DellEMC-Z9432f-O32
DellEMC-Z9432f-T64C64O8-DPB (Present)
```

 **NOTE:** If you configure the nondefault profile, the profile and the switch configuration are maintained when you upgrade Enterprise SONiC using the `image install` command — see [Software image management](#). If you upgrade the Enterprise SONiC image using the ONIE: Install OS option when the switch boots, the default port profile is used.

- To view the supported breakout interfaces (for example, in the DellEMC-Z9432f-T64C64O8-DPB profile), change to standard interface naming mode and enter the `show interface breakout modes` command.

```
admin@z9432f:~$ sonic-cl
sonic# config terminal
sonic(config)# interface-naming standard
sonic(config)# show interface breakout modes

Port Interface Supported Modes Default Mode

1/2 Eth1/2 1x400G, 2x200G, 2x100G, 4x100G, 8x50G, 8x25G, 8x10G 1x400G
1/4 Eth1/4 1x400G, 2x200G, 2x100G, 4x100G, 8x50G, 8x25G, 8x10G 1x400G
1/6 Eth1/6 1x400G, 2x200G, 2x100G, 4x100G, 8x50G, 8x25G, 8x10G 1x400G
```

|      |         |                                                        |        |
|------|---------|--------------------------------------------------------|--------|
| 1/8  | Eth1/8  | 1x400G, 2x200G, 2x100G, 4x100G, 8x50G,<br>8x25G, 8x10G | 1x400G |
| 1/10 | Eth1/10 | 1x400G, 2x200G, 2x100G, 4x100G, 8x50G,<br>8x25G, 8x10G | 1x400G |
| 1/12 | Eth1/12 | 1x400G, 2x200G, 2x100G, 4x100G, 8x50G,<br>8x25G, 8x10G | 1x400G |
| 1/14 | Eth1/14 | 1x400G, 2x200G, 2x100G, 4x100G, 8x50G,<br>8x25G, 8x10G | 1x400G |
| 1/16 | Eth1/16 | 1x400G, 2x200G, 2x100G, 4x100G, 8x50G,<br>8x25G, 8x10G | 1x400G |
| 1/17 | Eth1/17 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/18 | Eth1/18 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/19 | Eth1/19 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/20 | Eth1/20 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/21 | Eth1/21 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/22 | Eth1/22 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/23 | Eth1/23 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/24 | Eth1/24 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/25 | Eth1/25 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/26 | Eth1/26 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/27 | Eth1/27 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/28 | Eth1/28 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/29 | Eth1/29 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/30 | Eth1/30 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/31 | Eth1/31 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |
| 1/32 | Eth1/32 | 1x400G, 2x200G, 2x100G, 4x100G                         | 1x400G |

**i | NOTE:** 8x25 breakout interfaces are supported only on eight ports: Eth1/2 to Eth1/16.

- To split a Z9432F-ON port into breakout interfaces, enter the command:

```
sonic(config)# interface breakout port slot/port mode {1x100G | 1x40G | 2x100G |
2x50G | 4x100G | 4x25G | 4x10G | 1x400G | 2x200G | 4x50G | 8x50G | 8x25G | 8x10G}
```

To reset a port to its default mode, use the no interface breakout port slot/port mode command.

- To verify the configured breakout interfaces, use the show interface breakout command; for example:

```
sonic(config)# do show interface breakout

Port Breakout Mode Status Interfaces

1/2 8x25G Completed Eth1/2/1
 Eth1/2/2
 Eth1/2/3
 Eth1/2/4
 Eth1/2/5
 Eth1/2/6
 Eth1/2/7
 Eth1/2/8
1/8 8x25G Completed Eth1/8/1
 Eth1/8/2
 Eth1/8/3
 Eth1/8/4
 Eth1/8/5
 Eth1/8/6
 Eth1/8/7
 Eth1/8/8
1/12 8x25G Completed Eth1/12/1
 Eth1/12/2
 Eth1/12/3
 Eth1/12/4
 Eth1/12/5
 Eth1/12/6
 Eth1/12/7
 Eth1/12/8
```

# Port breakouts

**(i) NOTE:** Breakout ports are supported on S5212F-ON, S5224F-ON, S5232F-ON, S5248F-ON, S5296F-ON, S5448F-ON, Z9264F-ON, Z9332F-ON, Z9432F-ON, and Z9664F-ON switches. Breakout ports are not supported on N3248TE-ON, N3248PXE-ON, N3248X-ON, E3248PXE-ON, and E3248P-ON switches. To display the slot/ports and interfaces on each switch that support breakout ports, use the `show interface breakout modes` command.

Using a supported breakout cable, you can split a 40GE QSFP+, 100GE QSFP28, or 400GE QSFP56-DD Ethernet ports into separate breakout interfaces. All breakout interfaces must have the same speed. Dell Technologies recommends that you do not reconfigure the speed of a breakout interface.

```
sonic(config)# interface breakout port slot/port mode {1x100G | 1x40G | 2x100G | 2x50G | 4x100G | 4x25G | 4x10G | 1x400G | 2x200G | 4x50G | 8x50G | 8x25G | 8x10G}
```

- *slot/port* — Enter the physical port location
- 1x100G — Set a port to operate at 100G speed.
- 1x40G — Set a port to operate at 40G speed.
- 2x100G — Split a port into two 100G interfaces.
- 2x50G — Split a port into two 50G interfaces.
- 4x100G — Split a port into four 100G interfaces.
- 4x25G — Split a port into four 25G interfaces.
- 4x10G — Split a port into four 10G interfaces.
- 1x400G — Set a port to operate at 400G speed.
- 2x200G — Split a port into two 200G interfaces.
- 4x50G — Split a port into four 50G interfaces.
- 8x50G — Split a port into eight 50G interfaces.
- 8x25G — Split a port into eight 25G interfaces.
- 8x10G — Split a port into eight 10G interfaces.

For example:

```
sonic(config)# interface breakout port 1/21 mode 4x25G
```

If you configure an unsupported breakout mode, an error message displays:

```
sonic(config)# interface breakout port 1/3 mode 4x50G
%Error: Invalid or unsupported breakout mode 4x50G
```

Each breakout interface operates at the configured speed. Use the `no interface breakout port slot/port mode` command to reset a port to its default mode: 100G or 400G.

To configure an Ethernet breakout interface, use the `interface Eth slot/port[/breakout-port]` command.

## Before you break out a front-panel port

- Use the `show interface breakout modes` command to display the breakout modes supported on a switch. Here are examples of show output from different Enterprise SONiC switches:

```
sonic# show interface breakout modes

Port Pipe Interface Supported Modes Default Mode

1/1 1 Ethernet0 1x400G, 2x200G, 2x100G(8), 2x40G, 4x100G, 1x400G
 1x100G(4), 1x40G, 1x50G, 1x200G, 8x50G,
 8x25G, 8x10G, 4x25G, 4x10G, 1x25G, 1x10G
1/2 1 Ethernet8 1x400G, 2x200G, 2x100G(8), 2x40G, 4x100G, 1x400G
 1x100G(4), 1x40G, 1x50G, 1x200G, 8x50G,
 8x25G, 8x10G, 4x25G, 4x10G, 1x25G, 1x10G
1/3 1 Ethernet16 1x400G, 2x200G, 2x100G(8), 2x40G, 4x100G, 1x400G
 1x100G(4), 1x40G, 1x50G, 1x200G, 8x50G,
 8x25G, 8x10G, 4x25G, 4x10G, 1x25G, 1x10G
1/4 1 Ethernet24 1x400G, 2x200G, 2x100G(8), 2x40G, 4x100G, 1x400G
 1x100G(4), 1x40G, 1x50G, 1x200G, 8x50G,
 8x25G, 8x10G, 4x25G, 4x10G, 1x25G, 1x10G
```

```

1/5 2 Ethernet32 1x400G, 2x200G, 2x100G(8), 2x40G, 4x100G, 1x400G
 1x100G(4), 1x40G, 1x50G, 1x200G, 8x50G,
 8x25G, 8x10G, 4x25G, 4x10G, 1x25G, 1x10G
1/6 2 Ethernet40 1x400G, 2x200G, 2x100G(8), 2x40G, 4x100G, 1x400G
 1x100G(4), 1x40G, 1x50G, 1x200G, 8x50G,
 8x25G, 8x10G, 4x25G, 4x10G, 1x25G, 1x10G
...

```

```

sonic# show interface breakout modes

Port Pipe Interface Supported Modes Default Mode

1/1 N/A Eth1/1 1x100G, 1x40G, 2x50G, 1x50G, 4x25G, 1x100G
 4x10G
1/2 N/A Eth1/2 1x100G, 1x40G, 2x50G, 1x50G, 4x25G, 1x100G
 4x10G
1/3 N/A Eth1/3 1x100G, 1x40G, 2x50G, 1x50G, 4x25G, 1x100G
 4x10G
1/4 N/A Eth1/4 1x100G, 1x40G, 2x50G, 1x50G, 4x25G, 1x100G
 4x10G
1/5 N/A Eth1/5 1x100G, 1x40G, 2x50G, 1x50G, 4x25G, 1x100G
 4x10G
1/6 N/A Eth1/6 1x100G, 1x40G, 2x50G, 1x50G, 4x25G, 1x100G
 4x10G
...

```

```

sonic# show interface breakout modes

Port Pipe Interface Supported Modes Default Mode

1/1 4 Eth1/1 1x25G, 1x10G, 1x100G(4), 1x50G, 1x40G, 1x400G
 1x200G, 1x400G, 2x40G, 2x200G, 2x100G(8),
 4x25G, 4x10G, 4x100G
1/2 4 Eth1/2 1x25G, 1x10G, 1x100G(4), 1x50G, 1x40G, 1x400G
 1x200G, 1x400G, 2x40G, 2x200G, 2x100G(8),
 4x25G, 4x10G, 4x100G
1/3 1 Eth1/3 1x25G, 1x10G, 1x100G(4), 1x50G, 1x40G, 1x400G
 1x200G, 1x400G, 2x40G, 2x200G, 2x100G(8),
 4x25G, 4x10G, 4x100G
1/4 1 Eth1/4 1x25G, 1x10G, 1x100G(4), 1x50G, 1x40G, 1x400G
 1x200G, 1x400G, 2x40G, 2x200G, 2x100G(8),
 4x25G, 4x10G, 4x100G
1/5 1 Eth1/5 1x25G, 1x10G, 1x50G, 1x100G, 2x10G, 1x100G
 2x25G, 2x50G
1/6 1 Eth1/6 1x25G, 1x10G, 1x50G, 1x100G, 2x10G, 1x100G
 2x25G, 2x50G
...

```

- Use the `show interface breakout dependencies slot/port` command to verify if there are any existing configurations that are removed after you break out a front-panel port. For example, in this example, front-panel port 1/1 has the native interface name `Ethernet2` and is a member of VLAN 100.

```

sonic# show interface breakout dependencies port 1/1

Dependent Configurations

VLAN|Vlan100
VLAN_MEMBER|Vlan100|Ethernet2

```

- Use the `show interface breakout resources` command to display the maximum number of breakout ports supported per pipeline on the switch and the current consumption. For example, on a Z9664F-ON switch:

```

sonic# show interface breakout resources
Maximum ports supported in the system: 256
Current ports in the system: 64

Pipeline Ports Max-Ports Front-panel-ports

1 4 16 1/1, 1/2, 1/33, 1/34
10 4 16 1/19, 1/20, 1/51, 1/52
11 4 16 1/21, 1/22, 1/53, 1/54

```

|    |   |    |                        |
|----|---|----|------------------------|
| 12 | 4 | 16 | 1/23, 1/24, 1/55, 1/56 |
| 13 | 4 | 16 | 1/25, 1/26, 1/57, 1/58 |
| 14 | 4 | 16 | 1/27, 1/28, 1/59, 1/60 |
| 15 | 4 | 16 | 1/29, 1/30, 1/61, 1/62 |
| 16 | 4 | 16 | 1/31, 1/32, 1/63, 1/64 |
| 2  | 4 | 16 | 1/3, 1/4, 1/35, 1/36   |
| 3  | 4 | 16 | 1/5, 1/6, 1/37, 1/38   |
| 4  | 4 | 16 | 1/7, 1/8, 1/39, 1/40   |
| 5  | 4 | 16 | 1/9, 1/10, 1/41, 1/42  |
| 6  | 4 | 16 | 1/11, 1/12, 1/43, 1/44 |
| 7  | 4 | 16 | 1/13, 1/14, 1/45, 1/46 |
| 8  | 4 | 16 | 1/15, 1/16, 1/47, 1/48 |
| 9  | 4 | 16 | 1/17, 1/18, 1/49, 1/50 |

## Verify breakout configuration

```
sonic# show interface breakout

Port Breakout Mode Status Interfaces

1/21 4x25G Completed Eth1/21/1
 Eth1/21/2
 Eth1/21/3
 Eth1/21/4
```

## Display breakout port status

After you configure breakout interfaces, the interfaces are administratively and operationally down, such as Eth 1/3/1 to 1/3/4 in this example. To bring an interface administratively up, use the no shutdown command.

```
sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/1 - up oper-up off 100000 9100 Ethernet0
Eth1/2 - up oper-up off 100000 9100 Ethernet4
Eth1/3/1 - down admin-down off 25000 9100 Ethernet8
Eth1/3/2 - down admin-down off 25000 9100 Ethernet9
Eth1/3/3 - down admin-down off 25000 9100 Ethernet10
Eth1/3/4 - down admin-down off 25000 9100 Ethernet11

sonic(config)# interface range Eth 1/21/1-1/21/4
%Info: Configuring only existing interfaces in range
sonic(conf-if-range-eth**)# no shutdown

sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternate Name

Eth1/1 - up oper-up off 100000 9100 Ethernet0
Eth1/2 - up oper-up off 100000 9100 Ethernet4
Eth1/3/1 - up oper-up off 25000 9100 Ethernet8
Eth1/3/2 - up oper-up off 25000 9100 Ethernet9
Eth1/3/3 - up oper-up off 25000 9100 Ethernet10
Eth1/3/4 - up oper-up off 25000 9100 Ethernet11
```

## Port breakouts — usage notes

- There is no support to migrate interface configurations using a nondefault port profile in an Enterprise SONiC release before 3.1 to the default profile that supports port breakouts in release 3.1 or later. To reconfigure interfaces using the 3.1 or later default port profile, you must:
  - Back up the running configuration.
  - Install Enterprise SONiC 3.1 or later using ONIE and the onie-nos-install command.
  - Reconfigure the port breakouts using the interface breakout port mode command.
  - Reapply the backed-up running configuration.
- When you upgrade S5248F-ON and S5296F-ON switches to release 3.1 or later from an Enterprise SONiC release before 3.1, you must install SONiC using ONIE to receive breakout support on the 100G uplink ports. Also, you must follow the procedure in the previous bullet to migrate interface configurations. When you reapply the interface configurations, be sure to take into account the new interface numbering of 100G uplink ports in release 3.1 and later.

**Table 30. S5248F-ON 100G port numbering**

| <b>100G front panel port</b> | <b>Interface name in releases before 3.1</b> | <b>Interface name in release 3.1 and later</b> |
|------------------------------|----------------------------------------------|------------------------------------------------|
| 49                           | Ethernet48                                   | Ethernet48                                     |
| 51                           | Ethernet50                                   | Ethernet56                                     |
| 53                           | Ethernet52                                   | Ethernet64                                     |
| 54                           | Ethernet53                                   | Ethernet68                                     |
| 55                           | Ethernet54                                   | Ethernet72                                     |
| 56                           | Ethernet55                                   | Ethernet76                                     |

**Table 31. S5296F-ON 100G port numbering**

| <b>100G front panel port</b> | <b>Interface name in releases before 3.1</b> | <b>Interface name in release 3.1 and later</b> |
|------------------------------|----------------------------------------------|------------------------------------------------|
| 97                           | Ethernet96                                   | Ethernet96                                     |
| 98                           | Ethernet97                                   | Ethernet100                                    |
| 99                           | Ethernet98                                   | Ethernet104                                    |
| 100                          | Ethernet99                                   | Ethernet108                                    |
| 101                          | Ethernet100                                  | Ethernet112                                    |
| 102                          | Ethernet101                                  | Ethernet116                                    |
| 103                          | Ethernet102                                  | Ethernet120                                    |
| 104                          | Ethernet103                                  | Ethernet124                                    |

- In the Enterprise SONiC 3.1 and later release, port breakouts are not supported in nondefault port profiles.
- Dell Technologies recommends that you do not use the `interface breakout port mode` command and the `speed` command on the same interface.
- On a Z9332F-ON switch, Dell Technologies recommends that you use the one of the breakout modes to configure a port to run at 100G. The speed `100000` command is not supported.

```

sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# speed 100000
%Error: Unsupported speed

sonic(config)# interface breakout port 1/1 mode 2x100G
sonic(config)# interface Eth1/1/1
sonic(conf-if-Eth1/1/1)# no shutdown
sonic(conf-if-Eth1/1/1)# do show interface status | grep Eth1/1/1

Name Description Oper Reason AutoNeg Speed MTU Alternate Name
Eth1/1/1 - up oper-up off 100000 9100 Ethernet0

sonic(conf-if-Eth1/1)# do show interface breakout

Port Breakout Mode Status Interfaces

1/1/1 2x100G Completed Eth1/1/1
 Eth1/1/2

```

## Port channels

To provide redundancy, increased bandwidth, and better traffic load-balancing, bundle multiple physical interfaces into a single logical interface — a link aggregation group (LAG) or port channel. A port channel aggregates the bandwidth of member links. If a member port fails, traffic is redirected to the remaining ports.

A physical interface can belong to only one port channel at a time. A port channel must contain interfaces of the same speed. You can add up to 32 physical interfaces to a port channel. The configuration applied to the port channel bundle is used on all member interfaces. A port channel can operate in Active or On modes:

- Member interfaces in Active mode automatically initiate negotiations with other ports by using LACP packets, and set up a port channel with other ports in Active mode. Dynamic LACP is enabled on port interfaces in Active mode.
- Member interfaces in On mode are manually assigned as port-channel members. The LACP capability to dynamically group ports into a port channel is disabled.

## Port channel configuration

A port channel operates in either L2 (default) or L3 mode. To place a port channel in L3 mode and remove L2 configuration before you configure an IP address, use the `no switchport` command. To reconfigure a port channel in L2 mode, use the `switchport` command.

Perform these basic configuration tasks to set up port channels in your network.

### Create a port channel

```
sonic# interface PortChannel portchannel-number [mode {active | on}] [min-links number]
[fallback] [fast_rate]
```

For example:

```
sonic(config)# interface PortChannel 1 mode active min-links 2 fallback
sonic(config)# interface PortChannel 2 mode active fallback fast_rate
sonic(config)# interface PortChannel 3 mode on min-links 3
```

- Port-channel ID numbers are from 1 to 256.
- Configure the LACP mode for port channel members:
  - mode `active` — All interfaces in the port channel start-up with LACP enabled. Active ports dynamically negotiate with peer ports. Similarly configured ports are bundled as members of an active port channel. (Default)
  - mode `on` — Member ports operate as part of a static port channel with LACP disabled. A static port channel member transmits IP traffic if the port interface is up, but does not dynamically negotiate with peer ports.
- You can configure the minimum number of required links (1 to 255; default 0).
- By default, the admin status is UP, the MTU is 9100 bytes, fallback and fast rate are disabled, and the LACP mode is active.
  - The LACP fallback feature allows an active member interface to establish a connection with a peer interface before the port channel receives the LACP protocol negotiation from the peer.
  - When fast rate is disabled, an LACP port channel is in SLOW mode — Ethernet port members send LACP protocol data unit (LACP PDU) packets to connected neighbors with the state of the link every 30 seconds. If you enable the `fast_rate` option, the port channel operates in FAST mode — Ethernet members send LACP PDUs every second.

To delete a port channel, remove all member interfaces and enter the `no interface PortChannel portchannel-number` command.

### Add member interfaces

To create a static port channel LAG, manually assign member interfaces using the `channel-group` command.

- All members must be the same interface type (such as Ethernet).
- All member interfaces must have the same speed.
- An interface must not contain nondefault L2/L3 configuration settings. Only the `description` and `shutdown` or `no shutdown` commands are supported. You cannot add an IP address or static MAC address to a member interface.
- You can assign an interface to only one port channel.
- To remove a member interface, use the `no channel-group` command in Interface Configuration mode.

```
sonic(config)# interface Eth1/4
sonic(conf-if-Eth1/4)# channel-group 1
```

 **NOTE:** After you assign a Layer 2 port to a port channel, all switchport configurations must be done on the port channel. You can no longer apply switchport configurations to individual port channel members. Also, you cannot apply Layer 3 configurations to an individual port channel member either. You must apply configuration changes to the entire port channel.

### Configure a port channel

- **IP address** — Configure an IPv4 address on a port channel interface.
  - `ip address ip-address/mask` — Enter the IP address in dotted decimal format `A.B.C.D`. The `no ip address ip-address/mask` command removes the IP address on the interface.

### **IPv6 address** — Configure an IPv6 address on a port channel interface.

- `ipv6 address ipv6-address/prefix-length` — Enter a full 128-bit IPv6 address with the network prefix length, including the 64-bit interface identifier. You can configure multiple IPv6 addresses on an interface. The `no ipv6 address ipv6-address` command removes the IPv6 address on the interface.
  - To configure an IPv6 address besides the link-local address, use the `ipv6 address ipv6-address/prefix-length` command and specify the complete 128-bit IPv6 address.
  - To configure a globally unique IPv6 address by entering only the network prefix and length, use the `ipv6 address ipv6-address/prefix-length eui-64` command.

### **MTU** — Configure the maximum transmission unit (MTU) frame size for a port channel.

- `mtu value` — Enter the maximum frame size in bytes (1280 to 9216). The default port channel MTU is 9100 bytes. Enter `no mtu` to reset the default value.
- Configure the MTU on port channel members first before you configure the port channel MTU. All members of a port channel must have the same MTU value. Tagged members must have a link MTU 4 bytes higher than untagged members to account for the packet tag.
- Ensure that the MTU of port channel members is greater than or equal to the port channel MTU. If you configure the MTU on port channel members after you configure the port channel MTU, the port channel MTU may not be updated. The system selects the lowest MTU value that is configured on the port channel or port channel members to be the port channel MTU. For example, the port channel contains tagged members with Link MTU of 1522 and IP MTU of 1500 and untagged members with Link MTU of 1518 and IP MTU of 1500. The port channel's Link MTU cannot be higher than 1518 bytes and its IP MTU cannot be higher than 1500 bytes.
- **No shutdown** — Enable the port channel by entering `no shutdown`. To disable a port channel and place all member interfaces in an operationally down state, enter the `shutdown` command.

```
sonic# interface PortChannel 10
sonic(conf-if-po10)# mtu 2500
sonic(conf-if-po10)# ip address 2.2.2.2/24
sonic(conf-if-po10)# ipv6 address a::b/64
sonic(conf-if-po10)# no shutdown
```

### **Assign a port channel to a trunk or access VLAN**

A port channel can operate in either a trunk or access VLAN (see [Configure access and trunk interfaces](#)). Configure access and trunk operation in port channel configuration mode.

- An access port channel sends and receives untagged frames from connected L2 devices. The channel group interface is assigned to the access VLAN, which is VLAN 1 by default. To change the default access VLAN, use the `switchport access vlan vlan-id` command.
- A trunk port channel sends and receives tagged frames from multiple VLANs and untagged frames from the access VLAN. To configure a trunk port channel group, use the `switchport trunk allowed vlan vlan-id` command. To remove a tagged VLAN, enter the `no switchport trunk allowed vlan vlan-id` command.

To configure a port channel to operate in L2 Access or Trunk mode, use the `switchport` command. To restore a trunk port channel to L2 Access mode on VLAN1, use the `no switchport` command.

If you assign an IP address to a port channel, you cannot use the `switchport` command to enable L2 switching — you must first remove the IP address. Before you configure L3 mode on a port channel, use the `no switchport` command to remove all L2 configurations.

```
sonic(config)# interface PortChannel 4
sonic(conf-if-po4)# switchport access Vlan 5
```

```
sonic(config)# interface PortChannel 4
sonic(conf-if-po4)# switchport trunk allowed Vlan 5
```

### **Configure and delete a port channel**

```
Create static PortChannel 100
sonic(config)# interface PortChannel 100 mode on
sonic(conf-if-po100)# no shutdown
sonic(conf-if-po100)# exit

Add Ethernet interface to PortChannel
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# channel-group 100
sonic(conf-if-Eth1/1)# exit
```

```

#####
Add trunk and access VLANs to PortChannel
sonic(config)# interface PortChannel 100
sonic(conf-if-po100)# switchport trunk allowed Vlan add 1001
sonic(conf-if-po100)# switchport access Vlan 1
sonic(conf-if-po100)# end

#####
View interface members in PortChannel 100
sonic# show PortChannel summary
Flags(oper-status): D - Down U - Up (portchannel)
P - Up in portchannel (members)

Group PortChannel Type Protocol Member Ports

100 PortChannel100 (U) Eth NONE Eth1/1(P)

#####
Verify VLAN members in PortChannel 100
sonic# # show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
1 Active A PortChannel100 Enable No
1001 Active T PortChannel100 Enable No

#####
Step 1: Remove VLAN members from port channel
sonic# config terminal
sonic(config)# interface PortChannel 100
sonic(conf-if-po100)# no switchport trunk allowed Vlan 1001
sonic(conf-if-po100)# no switchport access Vlan
sonic(conf-if-po100)# do show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
1 Inactive Enable No
1001 Inactive Enable No
sonic(config)# exit

#####
Step 2: Remove interface members from port channel
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# no channel-group
sonic(config)# exit

#####
Step 3: Delete port channel
sonic(config)# no interface PortChannel 100
sonic(config)# do show PortChannel summary
Flags(oper-status): D - Down U - Up (portchannel)
P - Up in portchannel (members)

Group PortChannel Type Protocol Member Ports

```

sonic(config) #

### View port channel configuration

```

sonic# show PortChannel summary
Flags(oper-status): D - Down U - Up

Group PortChannel Type Protocol Member Ports

100 PortChannel100 (D) Eth LACP Eth1/1(U)
 Eth1/2(D)
200 PortChannel200 (D) Eth NONE Eth1/8(U)
300 PortChannel300 (D) Eth LACP Eth1/9(U)

```

```

sonic# show interface PortChannel
PortChannel1 is up, line protocol is down, mode Static
Hardware is PortChannel, address is 3c:2c:30:10:20:03
Minimum number of links to bring PortChannel up is 1
MTU 9100

```

```

PortChannel2 is up, line protocol is down, mode LACP
Hardware is PortChannel, address is 3c:2c:30:10:20:03
Minimum number of links to bring PortChannel up is 1
Fallback: Disabled

```

```

MTU 9100
LACP mode ACTIVE interval FAST priority 65535 address 90:b1:1c:f4:aa:b2

sonic# show interface PortChannel 91
PortChannel91 is up, line protocol is up, reason oper-up, mode LACP
Hardware is PortChannel, address is 54:bf:64:ba:43:c2
Minimum number of links to bring PortChannel up is 1
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Fallback: Disabled
Graceful shutdown: Disabled
MTU 9100
LineSpeed 10.0GB
Events:
iftrack-status-down at 2022-03-17T08:42:39.295531Z
iftrack-status-up at 2022-03-17T08:44:15.968168Z
delay-restore-status-up at 2022-03-17T08:44:18.62394Z
lacp-fail at 2022-03-17T08:44:19.495931Z
portchannel-up at 2022-03-17T08:44:19.682745Z
LACP mode ACTIVE interval SLOW priority 65535 address aa:bb:cc:dd:ee:ff
Members in this channel: Eth1/1(Selected)
LACP Actor port 1 address aa:bb:cc:dd:ee:ff key 91
LACP Partner port 129 address 3c:2c:30:09:b5:81 key 91
Last clearing of "show interface" counters: never
10 seconds input rate 500454 packets/sec, 4003637088 bits/sec, 500454636 Bytes/sec
10 seconds output rate 520440 packets/sec, 4173129256 bits/sec, 521641157 Bytes/sec
Input statistics:
 3669975749 packets, 3669770682224 octets
 508 Multicasts, 0 Broadcasts, 3669975260 Unicasts
 0 error, 82139932 discarded
Output statistics:
 3794709729 packets, 3803262211689 octets
 502 Multicasts, 0 Broadcasts, 3794709227 Unicasts
 0 error, 0 discarded
Time since last interface status change: 02:02:22

```

## Port channel reconfiguration

Starting in Release 4.1.0, the reconfiguration of port-channel parameters for existing port channels is supported "on the fly". It is no longer required to first delete the port channel, and then re-create and reconfigure it.

As shown in the example in [Port channel configuration](#), to reconfigure a port channel in releases earlier than 4.1.0, it was necessary to:

1. Remove all trunk and access VLAN members from a port channel.
2. Remove all interface members from port channel.
3. Delete the port channel.
4. Re-create the port channel with LACP enabled or disabled, and optional settings for minimum number of required links, LACP fallback, and fast rate.
5. Re-add Ethernet interfaces and trunk and access VLANs as members.

In Release 4.1.0 and later, you can reconfigure the port-channel parameters for minimum required links, fallback mode, and fast-rate in PortChannel Configuration mode; for example:

```

sonic(config)# interface PortChannel 4
sonic(conf-if-po4)# min-links 4
sonic(conf-if-po4)# fallback
sonic(conf-if-po4)# fast-rate

```

**(i) NOTE:** To reconfigure port-channel parameters "on the fly", you cannot use the `interface PortChannel portchannel-number` command in Configuration mode. An error message is displayed; for example:

```

sonic(config)# interface PortChannel 1 min-links 2
sonic(conf-if-po1)# exit
sonic(config)# interface PortChannel 1 min-links 4
%Error: Cannot reconfigure min links for an existing PortChannel: PortChannel1

```

To reset the port-channel parameters min-links, fallback mode, and fast-rate mode to their default values:

```
sonic(config)# interface PortChannel 4
sonic(conf-if-po4)# no min-links 4
sonic(conf-if-po4)# no fallback
sonic(conf-if-po4)# no fast-rate
```

To view "on the fly" changes in a port-channel configuration, use the `show interface PortChannel` command.

## Port-channel graceful shutdown

Use port-channel graceful shutdown when you perform a software upgrade on an MLAG peer switch — see [Multichassis LAG](#). Data connectivity is maintained by forwarding data through the MLAG peer. All port channels in the MLAG switch that is being upgraded are brought down operationally when you enable global port-channel graceful shutdown mode.

### Usage notes

- On a specified port channel:
  - If you enable graceful shutdown, the port channel is brought DOWN (operationally), regardless of the global port-channel graceful shutdown setting.
  - If you disable graceful shutdown, the port channel exits graceful shutdown mode and no longer shuts down with no frame loss.
- On all port channels on the switch:
  - If you enable graceful shutdown globally, all port channels are operationally brought DOWN, except for port channels that have graceful shutdown specifically disabled.
  - If you disable graceful shutdown globally, all port channels exit graceful shutdown mode and no longer shut down with no frame loss, except for port channels that have graceful shutdown specifically enabled.
- When you disable graceful shutdown and a port channel exits graceful shutdown mode:
  - The LACP state machine restarts on all port-channel member ports that are link up.
  - LACPDU s are transmitted and received on all port-channel member ports that are link up.
  - If LACP convergence succeeds, a port channel is operationally up.

### Enable port-channel graceful shutdown

- To enable port-channel graceful shutdown on a specified port channel:

```
sonic(config)# interface PortChannel 6
sonic(conf-if-po6)# graceful-shutdown
```

To disable graceful shutdown on a port channel:

```
sonic(config)# interface PortChannel 6
sonic(conf-if-po6)# no graceful-shutdown
```

- To enable graceful shutdown globally on all configured port channels on the switch:

```
sonic(config)# portchannel graceful-shutdown
```

To disable port-channel graceful shutdown globally:

```
sonic(config)# no portchannel graceful-shutdown
```

### View port-channel graceful shutdown

```
sonic# show interface PortChannel
PortChannel5 is up, line protocol is down, mode LACP
Minimum number of links to bring PortChannel up is 1
Fallback: Disabled
Graceful shutdown: Enabled
MTU 9100
LACP mode ACTIVE interval SLOW priority 65535 address 3c:2c:99:2d:81:35
Members in this channel: Ethernet220
LACP Actor port 221 address 3c:2c:99:2d:81:35 key 5
LACP Partner port 0 address 00:00:00:00:00:00 key 0
Last clearing of "show interface" counters: 1970-01-01 00:00:00
Input statistics:
```

```
 0 packets, 0 octets
0 Multicasts, 0 Broadcasts, 0 Unicasts
0 error, 0 discarded
Output statistics:
 0 packets, 0 octets
0 Multicasts, 0 Broadcasts, 0 Unicasts
0 error, 0 discarded
```

## VLANs

Virtual local area networks (VLANs) are logical interfaces that allow a group of devices to communicate as if they were in the same network, independent of physical location. By default, VLANs operate in L2 mode. Physical interfaces and port channels can be members of VLANs.

## VLAN configuration

Create a VLAN using the `interface vlan` command in global configuration mode, then configure basic settings. To add Ethernet and port channel member interfaces to an access or trunk VLAN, see [Configure access and trunk interfaces](#).

**i** **NOTE:** You can add an Ethernet or port channel interface to a nonexisting VLAN. During VLAN creation, all interfaces that are assigned to the VLAN ID are added as members.

```
sonic(config)# interface Vlan 10
sonic(conf-if-Vlan10) #
```

- VLAN IDs are from 1 to 4094; there is no default VLAN.
- To disable a VLAN, use the `no interface vlan vlan-id` command.

Perform these configuration tasks to set up VLANs in your network.

**IP address** — Configure an IPv4 address on a VLAN interface.

- `ip address ip-address/mask` — Enter the IP address in dotted decimal format *A.B.C.D*. The `no ip address ip-address/mask` command removes the IP address on the interface.

**IPv6 address** — Configure an IPv6 address on a VLAN interface.

- `ipv6 address ipv6-address/prefix-length` — Enter a full 128-bit IPv6 address with the network prefix length, including the 64-bit interface identifier. You can configure multiple IPv6 addresses on an interface. The `no ipv6 address ipv6-address` command removes an IPv6 address on the interface.

**MTU** — Configure the maximum transmission unit (MTU) frame size for IP traffic on VLAN member interfaces.

- `mtu value` — Enter the maximum frame size in bytes (1312 to 9216; default 9100 bytes). Enter `no mtu` to reset the default value. Configure the MTU on VLAN members first before you configure the VLAN MTU. All members of a VLAN must have the same MTU value. Tagged members must have a link MTU 4 bytes higher than untagged members to account for the packet tag.

### VLAN configuration

```
sonic(config)# interface Vlan 10
sonic(conf-if-Vlan10) # mtu 2500
sonic(conf-if-Vlan10) # ip address 2.2.2.2/24
sonic(conf-if-Vlan10) # ipv6 address a::b/64
```

### Assign interfaces as VLAN members

To assign an interface to a trunk or access VLAN, see [Configure access and trunk interfaces](#).

#### Configure and delete a VLAN

```
Add Ethernet interface to VLAN
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1) # switchport trunk allowed Vlan add 1001
sonic(conf-if-Eth1/1) # exit

Add port channel to VLAN

```

```

sonic(config)# interface PortChannel 10
sonic(conf-if-po10)# switchport trunk allowed Vlan add 1001
sonic(conf-if-po10)# exit

View VLAN members
sonic(config)# do show Vlan 1001
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
1001 Active T Eth1/1 Enable No
 PortChannel10

Step 1: Remove members from VLAN
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# no switchport access Vlan
sonic(conf-if-Eth1/1)# exit

sonic(config)# interface PortChannel 10
sonic(conf-if-po10)# no switchport trunk allowed Vlan 1001

sonic(conf-if-po100)# do show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
1001 Inactive Enable No
sonic(conf-if-po100)# exit

Step 2: Delete VLAN
sonic(config)# no interface Vlan 1001
sonic(config)# end
sonic# show Vlan

sonic#

```

### Display VLAN configuration

In show Vlan output:

- NUM — VLAN ID number
- Status — VLAN status displays as Active or Inactive.
  - Active — A VLAN member is present and the line protocol for at least one VLAN member is up.
  - Inactive — No VLAN member is present or the line protocol for all VLAN members is down.
- Q — Displays the 802.1Q mode of a VLAN member interface:
  - T — Tagged VLAN member
  - A — Access VLAN member
- Autostate — Displays the VLAN autostate mode: Enable or Disable.
- Dynamic — Yes indicates a RADIUS-supplied VLAN (see **RADIUS-supplied VLANs** in [Port access control](#)); No indicates a static VLAN that was manually configured.

```

sonic# show Vlan
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
 5 Active T Eth1/4 Enable Yes
 10 Inactive Enable No
 20 Inactive A PortChannel10 Enable No

```

```

sonic# show Vlan 5
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
 5 Active T Eth1/4 Enable Yes
 T PortChannel10
 A Eth1/3

```

```

sonic# show interface Vlan 101
Vlan101 is up, line protocol is up
Hardware is Vlan, address is 20:04:0f:46:b7:bb
IPV4 address is 172.11.0.1/24
Mode of IPV4 address assignment: MANUAL
IPV6 address is 2005:0:0:1::1/64, fe80::2005:1ff:fe46:b7bb/64
Mode of IPV6 address assignment: MANUAL

```

```
IP MTU 9100 bytes
Time since last interface status change: 21:56:22
```

## VLAN autostate

VLAN autostate determines the operational status of a VLAN.

### Enable VLAN autostate

VLAN autostate is enabled by default on all VLANs.

When enabled, the operational status of a VLAN is determined according to the operational status of its physical port and port channel members:

- **Active** — A VLAN member is present and the line protocol for at least one VLAN member is up.
- **Inactive** — No VLAN member is present or the line protocol for all VLAN members is down.

In a VXLAN, symmetric integrated bridging and routing (IRB) uses a VRF-VNI mapping for an IRB VLAN - see [Configure symmetric IRB](#). When autostate is enabled, if you configure a VRF-VNI mapping and a VLAN-VNI mapping for the IRB VLAN, the VLAN is declared operationally up irrespective of the operational state of its members.

For example, if host VLAN 100 is mapped to a VXLAN ID (VNI 1000) and a tenant VRF (Vrf-Red) is mapped to VNI 1000, if autostate is enabled, host VLAN 100 is considered to be operationally up.

### Disable VLAN autostate

In certain switch deployments, a VLAN must be considered operationally up (**Active**) even if all member interfaces are operationally down. A VLAN's operational state must be up (**Active**) in order for L3 protocols to operate over the VLAN interface even if there are no physical port or port-channel members.

For example, in a VXLAN that uses asymmetric IRB, it is desirable that L3 forwarding to remote hosts connected over the VXLAN tunnel continues unhindered without taking into account the presence or operational state of the local VLAN members — see [Configure asymmetric IRB](#). To ensure this VXLAN operation, disable autostate on the VLAN extended over the tunnel.

### Configure VLAN autostate mode

To disable VLAN autostate:

```
sonic(conf-if-Vlan100)# no autostate
sonic(conf-if-range-vl***)# no autostate
```

To re-enable VLAN autostate:

```
sonic(conf-if-Vlan100)# autostate
sonic(conf-if-range-vl***)# autostate
```

### View VLAN autostate mode

```
sonic# show Vlan
Q: A - Access (Untagged), T - Tagged
 NUM Status Q Ports Autostate Dynamic
 100 Inactive T Ethernet0 Enable Yes
 T Ethernet1
 101 Active T Ethernet1 Disable No
 108 Active T PortChannel101 Disable No
 200 Inactive T Ethernet1 Enable No
 201 Inactive T Ethernet1 Enable No
 202 Inactive T Ethernet1 Enable No
 203 Active T Ethernet1 Disable No

sonic# show Vlan 101
Q: A - Access (Untagged), T - Tagged
 NUM Status Q Ports Autostate Dynamic
 101 Active T Ethernet1 Disable No
```

**(i) NOTE:** In the show running config output for a VLAN, only the autostate disable mode is displayed.

```
sonic# show running-configuration interface Vlan 100
interface Vlan100
 no autostate
```

## Q-in-Q VLAN tunneling

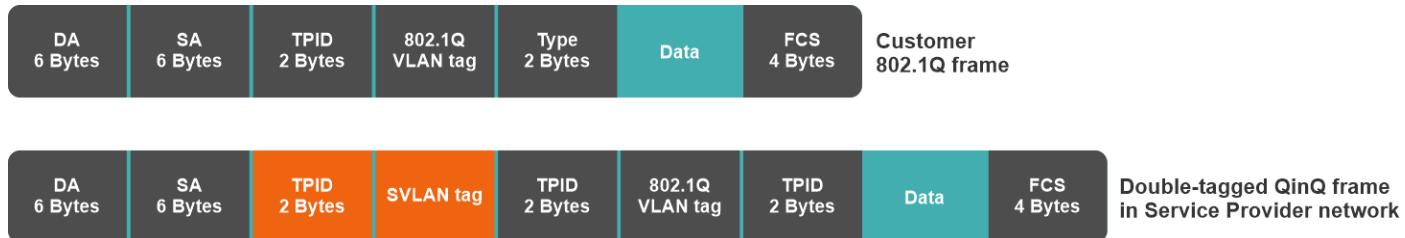
Use Q-in-Q VLAN tunneling to configure how customer VLAN traffic is transmitted over a service provider network.

Q-in-Q VLAN tunneling allows service providers to separate VLAN traffic from different customers by tunneling multiple VLANs from one customer (CVLANs) in a single, customer-specific service-provider VLAN (SVLAN). When the customer traffic enters the service provider network, a second 802.1Q tag is added to a customer-tagged frame. The encapsulated packet consists of an inner CVLAN tag of the private customer network and an outer SVLAN tag of the public provider network.

Q-in-Q VLAN tunneling is supported on physical interfaces — Ethernet and port channels. Q-in-Q tunneling is also known as VLAN stacking and complies with the 802.1ad standard.

If the provider network uses a VXLAN overlay (see [VXLAN](#)), the customer VLAN traffic that is identified by an SVLAN is mapped to a VXLAN network identifier (VNI) and forwarded based on the VNI. The SVLAN header is replaced with a VNI. Each unique VNI maintains L2 isolation (separate bridging domains) from other customer tenant segments. Before egress to a customer network, the VNI in VLAN packets is mapped to the SVLAN. Forwarding decisions are made by a Provider Edge switch based on the SVLAN.

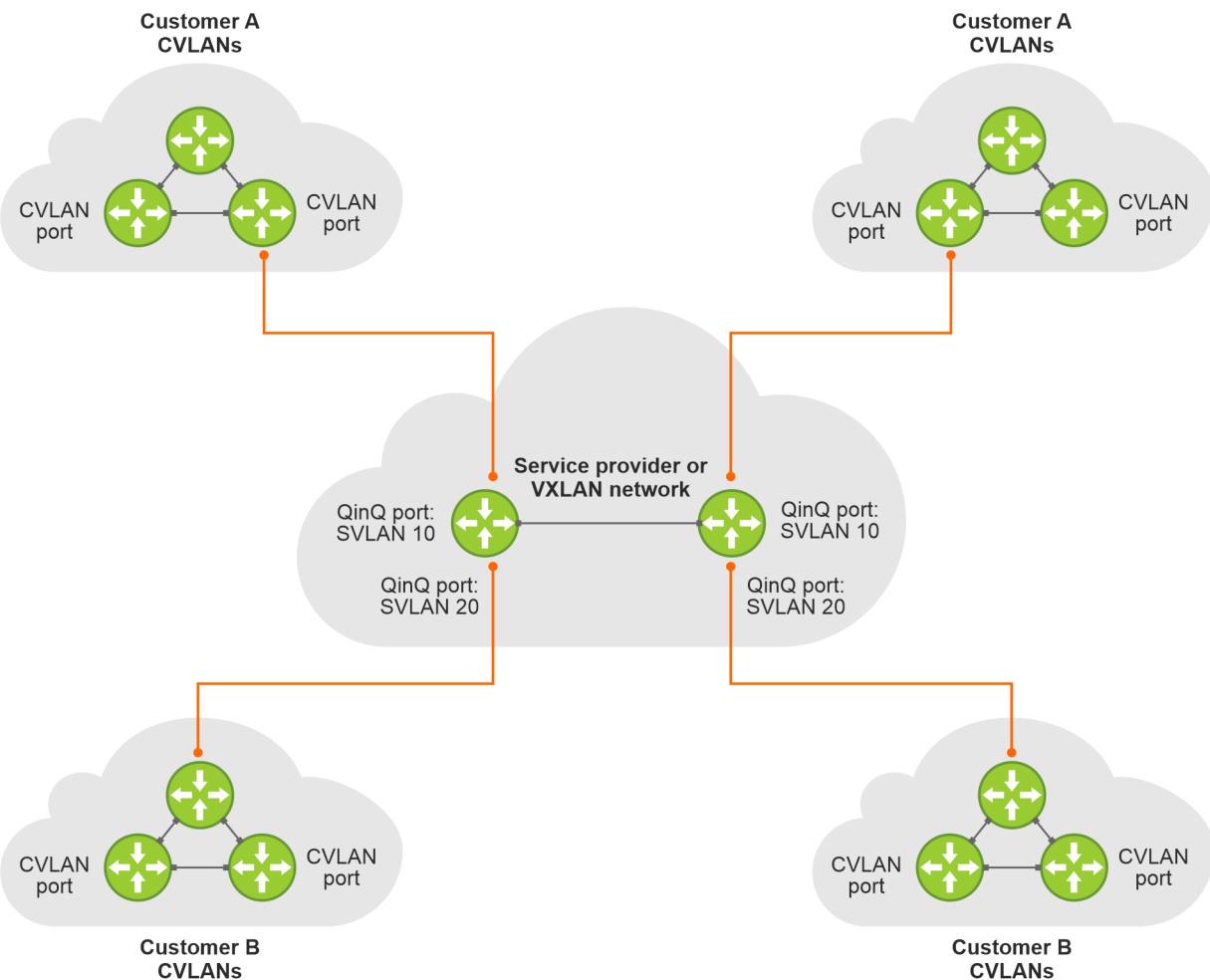
Using a double 802.1Q tag format, Q-in-Q packets have four more bytes than single 802.1Q tagged VLAN packets:



**Figure 1. CVLAN and Q-in-Q frame headers**

In Q-in-Q tunneling, customer traffic arriving at a service Provider Edge (PE) interface is encapsulated with an SVLAN tag that identifies a single CVLAN or a group of CVLANs. The packets are transmitted through the provider network based on the SVLAN ID while preserving the original customer VLAN ID in the packet. Although the packet is double-tagged, the forwarding decisions are made based on the outer SVLAN ID. At the egress of the provider network, when the traffic leaves a PE switch, the SVLAN in the packet is removed before forwarding the traffic to the customer network.

For example, in the following Q-in-Q figure, customer A VLAN traffic is encapsulated in SVLAN 10 and transmitted over the provider network. Customer B VLAN traffic is encapsulated in SVLAN 20 and transmitted over the provider network. If the service provider network uses a VXLAN, the SVLAN IDs for SVLAN 10 and SVLAN 20 would be replaced by a unique VXLAN network identifier for each customer segment and forwarded across the provider network.



**Figure 2. Q-in-Q tunneling in service provider network**

#### Q-in-Q VLAN restrictions

- Q-in-Q tunneling is supported only on Trident3 (TD3) switches — Z9332F-ON, S5200F-ON series, E3200F-ON series, and N3200F-ON series — and Trident4 (TD4) switches — Z9432F-ON and S5448F-ON — with the Z9432F-ON and S5448F-ON restrictions described in the next section.
- Q-in-Q tunneling is not supported on MLAG port-channel interfaces.
- You cannot configure the same SVLAN for both Q-in-Q VLAN tunneling and VLAN translation on an interface.
- You can configure trunk or access ports as members of any other VLANs on an interface if a VLAN is not used as an SVLAN.
- Only Layer 2 traffic is supported on an SVLAN; Layer 3 configuration is not supported.
- STP and IGMP snooping are not supported on an SVLAN used for Q-in-Q VLAN tunneling.
- Only TPID 0x8100 is supported in Q-in-Q frame headers.

#### Z9432F-ON and S5448F-ON restrictions

By default, VLAN translation is not enabled on Z9432F-ON and S5448F-ON switches. You must first enable it before you can configure VLAN translation as described in [Configure VLAN translation](#). To enable these switches for VLAN stacking, enter the `vlan-stacking` command from the `switch-resource` command tree. Then save the configuration and reload the switch.

```
sonic(config)# switch-resource
sonic(config-switch-resource)# vlan-stacking
Config save and reboot is required for this change to take effect
sonic(config-switch-resource)# exit
sonic# write memory
sonic# reload
```

After the switch reboots, verify that VLAN stacking is enabled by entering the `show switch-resource vlan-stacking` command:

```
sonic# show switch-resource vlan-stacking
Configured : enabled
Operational : enabled
```

**(i) NOTE:** If you check the VLAN stacking status before you save and reload the switch, it is still operationally disabled:

```
sonic(config)# switch-resource
sonic(config-switch-resource)# vlan-stacking
Config save and reboot is required for this change to take effect
sonic(config-switch-resource)# do show switch-resource vlan-stacking
Configured : enabled
Operational : disabled
```

**(i) NOTE:** To disable VLAN stacking features on the specified switches, enter the `no vlan-stacking` command:

```
sonic(config)# switch-resource
sonic(config-switch-resource)# no vlan-stacking
Config save and reboot is required for this change to take effect
sonic(config-switch-resource)# exit
sonic# write memory
sonic# reload

! After reboot:
sonic# show switch-resource vlan-stacking
Configured : disabled
Operational : disabled
```

## Configure Q-in-Q VLAN tunneling

1. On a customer edge switch, configure each PE-facing interface to operate as a trunk member of an SVLAN:

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# switchport vlan-mapping {cvlan-list | {add | remove} cvlan-list} dot1q-tunnel svlan-id [priority priority-bits]
```

- *cvlan-list* — Enter the customer VLAN IDs to be mapped to an SVLAN, or to be added or removed from the list of CVLANs already mapped to the SVLAN. Enter a single CVLAN ID or a range with a hyphen (-); for example, 10-21. Separate CVLAN IDs and ranges using a comma (,); for example, 10,15-18,21.
- *{add | remove} cvlan-list* — Specify the customer VLAN IDs to be added or removed from the list of CVLAN traffic that is mapped to the specified SVLAN.
- *svlan-id* — Specify the service-provider VLAN ID used to transmit CVLAN traffic (1-4094). You cannot configure the same SVLAN ID in a Q-in-Q tunnel configuration that has already been used in a CVLAN to SVLAN mapping — see [VLAN translation](#). Only L2 traffic is supported on an SVLAN; L3 configuration settings are not supported.
- *priority priority-bits* — (Optional) Set the priority bits in the SVLAN tag (0-7).

For information about how to configure a trunk port, see [Configure access and trunk interfaces](#).

**(i) NOTE:** To delete a Q-in-Q VLAN, enter:

```
sonic(conf-if-Eth1/1)# switchport vlan-mapping dot1q-tunnel svlan-id [priority priority-bits]
```

2. On a PE switch interface, configure the Q-in-Q VLAN tunnel for a list of CVLANs. Optionally, edit the CVLAN list by adding or removing CVLAN entries. The configured priority bit is used in SVLAN frames in the provider network. If you do not specify a priority bit, the SVLAN inherits the priority bits in CVLAN tags. If you specify an SVLAN priority bit, it takes precedence over CVLAN priority bits in both ingress and egress directions in traffic on PE access ports.

```
sonic(conf-if-Ethslot/port)# switchport vlan-mapping {cvlan-list | {add | remove} cvlan-list} dot1q-tunnel svlan-id [priority priority-bits]
```

For example:

```
sonic(conf-if-Eth1/1)# switchport vlan-mapping 20,30 dot1q-tunnel 100 priority 3
sonic(conf-if-Eth1/1)# switchport vlan-mapping add 30-40 dot1q-tunnel 100
sonic(conf-if-Eth1/1)# switchport vlan-mapping remove 39 dot1q-tunnel 100

sonic(conf-if-po100)# switchport vlan-mapping 20,30 dot1q-tunnel 100 priority 3
sonic(conf-if-po100)# switchport vlan-mapping add 30-40 dot1q-tunnel 100
sonic(conf-if-po100)# switchport vlan-mapping remove 39 dot1q-tunnel 100
```

To remove a Q-in-Q VLAN tunnel configuration, enter the `no switchport vlan-mapping svlan-id` command.

3. (Optional) If the provider network uses a VXLAN overlay, follow these steps:

- Map the customer VLAN traffic that is identified by an SVLAN to a VXLAN network identifier (VNI) on a VTEP; for example:

```
sonic(config)# interface vxlan vtep-stacking
sonic(config-if-vxlan-vtep-stacking)# map vni 10010 vlan 100
```

To verify the VXLAN VNI to SVLAN mapping:

```
sonic# show vlan 100
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
100 Active T Ethernet69 Enable No
 A Vxlan_44.44.44.44 No

sonic# show interface vlan-mappings dot1q-tunnel

Name Vlan dot1q-tunnel Vlan Priority

Ethernet69 1250-1277 100 5
Ethernet69 1300-1310 2200 -
PortChannel99 1581-1590 1625 3
PortChannel99 1551-1575 1650 -
```

To verify the VXLAN VNI-to-SVLAN mapping:

```
sonic# show vxlan vlanvnimap | grep 100
Vlan100 10010
```

- Configure the BGP route-target settings; for example:

```
sonic(config)# interface Ethernet 121
sonic(conf-if-Ethernet121)# mtu 9100
sonic(conf-if-Ethernet121)# speed 10000
sonic(conf-if-Ethernet121)# no shutdown
sonic(conf-if-Ethernet121)# switchport trunk allowed Vlan 1-512
sonic(conf-if-Ethernet121)# switchport vlan-mapping 550 3101 priority 2
sonic(conf-if-Ethernet121)# switchport vlan-mapping 600 inner 700 3102 priority 5
sonic(conf-if-Ethernet121)# switchport vlan-mapping 701-800 dot1q-tunnel 3103
priority 7
```

To verify the BGP route-target settings:

```
sonic# show Vlan 3101
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
3101 Active T Ethernet120 Enable No
 T Ethernet121 No
 A Vxlan_13.13.13.13 No
```

On leaf nodes:

```
sonic# show running-configuration bgp vrf default
address-family l2vpn evpn
!
vni 203103
 route-target both auto
```

```
sonic# show running-configuration bgp vrf default
address-family l2vpn evpn
!
vni 303103
route-target both auto
```

On border leaf nodes:

```
sonic# show running-configuration bgp vrf default
address-family l2vpn evpn
!
vni 203103
route-target both auto
route-target import 303103:3103
route-target export 203103:3103
!

sonic# show running-configuration bgp vrf default
address-family l2vpn evpn
!
vni 203103
route-target both auto
route-target export 303103:3103
route-target import 203103:3103
```

## View Q-in-Q VLAN tunneling

To display the Q-in-Q VLAN configurations on all interfaces:

```
sonic# show interface vlan-mappings dot1q-tunnel

Name Vlan dot1q-tunnel Vlan Priority

Eth1/2 10 100 -
Eth1/2 11-20 200 7
Eth1/4 30,32,35-40 300 2
PortChannel10 41 400 -
```

To display the Q-in-Q VLAN configurations on a specified interface:

```
sonic# show interface Eth1/2 vlan-mappings dot1q-tunnel

Name Vlan dot1q-tunnel Vlan Priority

Eth1/2 10 100 -
Eth1/2 11-20 200 7

sonic# show running-configuration interface Eth1/2
!
interface Eth1/2
 mtu 9100
 speed 10000
 no shutdown
 switchport vlan-mapping 10 dot1q-tunnel 100
 switchport vlan-mapping 11-20 dot1q-tunnel 200 priority 7
```

```
sonic# show running-configuration interface PortChannel 10
!
interface PortChannel 100
 switchport vlan-mapping 41 dot1q-tunnel 400
 no shutdown
```

# VLAN translation

Use VLAN translation to configure how customer VLAN traffic is transmitted over a service provider network.

VLAN translation differs from Q-in-Q tunneling in that single- or double-tagged customer VLAN traffic (CVLANs) is swapped with an SVLAN at the ingress provider edge device. The traffic is then forwarded based on the SVLAN ID in the provider network. The CVLANs are lost in the translation. VLAN translation for provider networks is supported on Ethernet and port-channel interfaces.

If the provider network uses a VXLAN overlay (see [VXLAN](#)), the customer VLAN traffic that is identified by an SVLAN is mapped to a VXLAN network identifier (VNI) and forwarded based on the VNI. The SVLAN header is replaced with a VNI. Each unique VNI maintains L2 isolation (separate bridging domains) from other customer tenant segments. Before egress to a customer network, the VNI in VLAN packets is mapped to the SVLAN. Forwarding decisions are made by a Provider Edge switch based on the SVLAN.

## VLAN translation restrictions

- You cannot configure an SVLAN for both Q-in-Q VLAN tunneling and VLAN translation on an interface.
- VLAN translation is not supported on MLAG port-channel interfaces.
- On an interface, you can configure trunk or access ports as members of any VLANs that are not used as an SVLAN.
- Only Layer 2 traffic is supported on an SVLAN; Layer 3 configuration is not supported.
- In `switchport vlan-mapping` commands, the `multi-tag` option is supported only on Trident3 switches — Z9332F-ON, S5200F-ON series, E3200F-ON series, and N3200F-ON series — for an SVLAN mapped to a VXLAN VNI. The SVLAN can be mapped to either a single- or double-tagged CVLAN but not both for multi-tag support. The SVLAN cannot be configured as an access or trunk VLAN.
- On a physical port or port-channel interface, you can configure multiple VLAN translations associated with the same SVLAN only on Trident3 switches.
- When you configure VLAN translation (CVLAN-to-SVLAN mapping) for a single- or double-tagged customer VLAN or a single-tagged with multi-tag or double-tagged with multi-tag customer VLAN: the HW/SAI supports a maximum of 4000 rules (ingress and egress), which is equivalent to 2000 service chains (Q-in-Q or VLAN translation) per switch. In VLAN translation and Q-in-Q VLANs, one ingress and one egress rule constitute a service chain on a port. There are different ways in which the maximum limit of 4000 HW rules is reached. For example, the following CVLAN-to-SVLAN translations on port Eth1/5/1 consume different amounts of HW rules out of the maximum 4000 rules that are supported:

```
sonic(config)# interface Eth1/5/1
!
! Configure VLAN translation with one ingress and one egress rule that consumes 2 out of
! 4000 HW rules; applicable to a single CVLAN or multiple CVLANs with or without multi-tag
sonic(conf-if-Eth1/5/1)# switchport vlan-mapping 100 1000

!
! Configure Q-in-Q VLAN tunneling with ten ingress and one egress rule that consumes 11 out
! of 4000 HW rules; applicable to a single CVLAN, a list of CVLANs, or a CVLAN range
sonic(conf-if-Eth1/5/1)# switchport vlan-mapping 11-20 dot1q-tunnel 2000
```

## Z9432F-ON and S5448F-ON restrictions

By default, VLAN stacking is not enabled on Z9432F-ON and S5448F-ON switches. You must first enable it before you can configure VLAN translation as described in [Configure VLAN translation](#). To enable these switches for VLAN stacking, enter the `vlan-stacking` command from the `switch-resource` command tree. Then save the configuration and reload the switch.

```
sonic(config)# switch-resource
sonic(config-switch-resource)# vlan-stacking
Config save and reboot is required for this change to take effect
sonic(config-switch-resource)# exit
sonic# write memory
sonic# reload
```

After the switch reboots, verify that VLAN stacking is enabled by entering the `show switch-resource vlan-stacking` command:

```
sonic# show switch-resource vlan-stacking
Configured : enabled
Operational : enabled
```

**(i) NOTE:** If you check the VLAN stacking status before you save and reload the switch, it is still operationally disabled:

```
sonic(config)# switch-resource
sonic(config-switch-resource)# vlan-stacking
Config save and reboot is required for this change to take effect
sonic(config-switch-resource)# do show switch-resource vlan-stacking
Configured : enabled
Operational : disabled
```

**(i) NOTE:** To disable VLAN stacking features on the specified switches, enter the no vlan-stacking command:

```
sonic(config)# switch-resource
sonic(config-switch-resource)# no vlan-stacking
Config save and reboot is required for this change to take effect
sonic(config-switch-resource)# exit
sonic# write memory
sonic# reload

! After reboot:
sonic# show switch-resource vlan-stacking
Configured : disabled
Operational : disabled
```

## Configure VLAN translation

1. On a PE switch interface, configure the CVLAN-to-SVLAN mapping for a single- or double-tagged customer VLAN. Re-enter the command to configure an SVLAN for more than one VLAN translation (single- or double-tagged) on an interface. For a single-tagged VLAN translation:

```
sonic(conf-if-Ethslot/port)# switchport vlan-mapping cvlan-id svlan-id [priority priority-bits] [multi-tag]
```

For a double-tagged customer VLAN translation:

```
sonic(conf-if-Ethslot/port)# switchport vlan-mapping cvlan-id [inner inner-cvlan-id] svlan-id [priority priority-bits] [multi-tag]
```

- *cvlan-id* — Matches a customer VLAN ID (1-4094). *cvlan-id* is a mandatory parameter for both single- and double-tagged VLAN packets.
- *inner cvlan-id* — Matches a customer VLAN ID (1-4094) for double-tagged packets. The outer VLAN tag is identified by the preceding *cvlan-id* parameter.
- *svlan-id* — Specifies the service-provider VLAN ID used to transmit customer VLAN traffic (1-4094). You cannot configure the same SVLAN ID for CVLAN-to-SVLAN mapping that has already been used in a Q-in-Q (dot1q) tunnel configuration — see [Q-in-Q VLAN tunneling](#).
- *priority priority-bits* — (Optional) Sets the priority bit in the SVLAN tag (0-7). The configured priority bit is used in SVLAN frames in the provider network. If you do not specify a priority bit, the SVLAN inherits the priority bits in CVLAN tags. If you specify an SVLAN priority bit, it takes precedence over CVLAN priority bits in both ingress and egress directions in traffic on PE access ports.
- *multi-tag* — (Optional) Allows unknown customer VLAN tags to be sent as a payload across a VxLAN network. An unknown customer VLAN tag can be part of a traffic flow for single- or double-tagged CVLAN-to-SVLAN translation. The *multi-tag* option is only supported on flows that are configured for VXLAN. If you configure a multi-tag CVLAN-to-SVLAN mapping, all flows on the SVLAN must be either all single-tagged or all double-tagged CVLANs.

For example, a single-tagged CVLAN 100 to SVLAN 200 translation:

```
sonic(conf-if-Eth1/1)# switchport vlan-mapping 100 200 priority 3
```

A double-tagged VLAN packet with an outer CVLAN 100 and inner dot1q 200 to SVLAN 300 translation:

```
sonic(conf-if-Eth1/1)# switchport vlan-mapping 100 inner 200 300
```

**i** **NOTE:** You can configure the same SVLAN ID to map both single-tagged and double-tagged CVLAN IDs; for example:

```
sonic(conf-if-Eth1/1) # switchport vlan-mapping 10 100
sonic(conf-if-Eth1/1) # switchport vlan-mapping 10 inner 2 100
sonic(conf-if-Eth1/1) # switchport vlan-mapping 10 inner 3 100
```

**i** **NOTE:** To remove all traffic flows for a configured SVLAN translation, enter the `no switchport vlan-mapping svlan-id` command.

```
sonic(conf-if-Eth1/1) # switchport vlan-mapping 150 350 priority 3 multi-tag
sonic(conf-if-Eth1/1) # no switchport vlan-mapping 350
```

To remove the CVLAN-to-SVLAN mapping for a specified flow, enter the `no switchport vlan-mapping cvlan-id [inner inner-cvlan-id] svlan-id` command.

```
sonic(conf-if-Eth1/1) # no switchport vlan-mapping 101 200
sonic(conf-if-Eth1/2) # no switchport vlan-mapping 360 inner 160 460
```

To set and delete the priority on a per-flow basis for specified single- or double-tagged CVLAN-to-SVLAN mapping, add the `priority` parameter:

```
sonic(conf-if-Eth1/3) # switchport vlan-mapping 60 1000 priority 6
sonic(conf-if-Eth1/3) # no switchport vlan-mapping 60 1000 priority

sonic(conf-if-Eth1/3) # switchport vlan-mapping 50 inner 20 1000 priority 4
sonic(conf-if-Eth1/3) # no switchport vlan-mapping 50 inner 20 1000 priority
```

2. (Optional) If the provider network uses a VXLAN overlay, follow these steps:

- Map the customer VLAN traffic that is identified by an SVLAN to a VXLAN network identifier (VNI) on a VTEP; for example:

```
sonic(config)# interface vxlan vtep-stacking
sonic(config-if-vxlan-vtep-stacking)# map vni 10020 vlan 200
```

To verify the VXLAN VNI to SVLAN mapping:

```
sonic# show vxlan vlanvnimap | grep 200
Vlan200 10020
```

- Configure the BGP route-target settings; for example:

```
sonic(config)# interface Ethernet 121
sonic(conf-if-Ethernet121) # mtu 9100
sonic(conf-if-Ethernet121) # speed 10000
sonic(conf-if-Ethernet121) # no shutdown
sonic(conf-if-Ethernet121) # switchport trunk allowed Vlan 1-512
sonic(conf-if-Ethernet121) # switchport vlan-mapping 550 3101 priority 2
sonic(conf-if-Ethernet121) # switchport vlan-mapping 600 inner 700 3102 priority 5
sonic(conf-if-Ethernet121) # switchport vlan-mapping 701-800 dot1q-tunnel 3103
priority 7
```

To verify the BGP route-target settings:

```
sonic# show Vlan 3101
Q: A - Access (Untagged), T - Tagged
NUM Status Q Ports Autostate Dynamic
3101 Active T Ethernet120 Enable No
 T Ethernet121
 A Vxlan_13.13.13.13 No

sonic# show running-configuration interface vxlan | grep 3101
map vni 203101 vlan 3101

sonic# show running-configuration bgp vrf default | find 203101
vni 203101
```

```

 route-target both auto
!
vni 203102
 route-target both auto
!
vni 203103
 route-target both auto

```

## View VLAN translation

To display all CVLAN to SVLAN mappings configured on the switch:

```

sonic# show interface vlan-mappings
Flags: M - Multi-tag

Name Outer Inner Mapped Vlan Priority Flags

Eth1/1 100 - 1000 - M
Eth1/1 200 20 2000 3 -
Eth1/7 100 - 1000 - M
PortChannel10 400 40 3000 - -

```

**i | NOTE:** In the Flags column, M indicates multi-tag CVLAN-to-SVLAN mapping, and is only displayed for Trident3 switches — Z9332F-ON, S5200F-ON series, E3200F-ON series, and N3200F-ON series — that support the multi-tag option.

To display all CVLAN to SVLAN mappings configured on an Ethernet or port-channel interface:

```

sonic# show interface Eth1/1 vlan-mappings
Flags: M - Multi-tag

Name Outer Inner Mapped Vlan Priority Flags

Eth1/1 100 - 1000 - M
Eth1/1 200 20 2000 0 -

```

```

sonic# show running-configuration interface PortChannel 10
interface PortChannel 10
 switchport vlan-mapping 41 dot1q-tunnel 400 priority 0
 switchport vlan-mapping 400 inner 40 3000
 no shutdown

```

## Configure VLAN translation using REST API

To configure VLAN translation for single- and double-tagged CVLANs, you can use REST operations.

### Example: Configure and delete single-tagged CVLAN-to-SVLAN translation using REST API

```

curl -X PATCH "https://100.94.113.29/restconf/data/openconfig-interfaces:interfaces/
interface=Ethernet8/openconfig-interfaces-ext:mapped-vlans" -H "accept: */*" -H
"Authorization: Basic YWRtaW46YWRtaW4xMjM=" -H "Content-Type: application/
yang-data+json" -d "{\"openconfig-interfaces-ext:mapped-vlans\":{\"mapped-vlan\":
[{\\"vlan-id\":100,\\"config\":{\\"vlan-id\":100},\\"match\":{\\"match-single-tags\":{\\"match-
single-tag\":[{\\"outer-vlan\":70,\\"config\":{\\"outer-vlan\":70,\\"priority\":3}},{\\"outer-
vlan\":80,\\"config\":{\\"outer-vlan\":80,\\"priority\":2}},{\\"outer-vlan\":90,\\"config\":
{\\"outer-vlan\":90}}]}},\\"ingress-mapping\":{\\"config\":{\\"vlan-stack-action\":
\"SWAP\"}},\\"egress-mapping\":{\\"config\":{\\"vlan-stack-action\\":\"SWAP\"}}}]}}"
curl -X DELETE "https://100.94.113.29/
restconf/data/openconfig-interfaces:interfaces/interface=Ethernet8/openconfig-interfaces-
ext:mapped-vlans/mapped-vlan=100/match/match-single-tags/match-single-tag=80" -H
"accept: */*" -H "Authorization: Basic YWRtaW46YWRtaW4xMjM="
curl -X DELETE "https://100.94.114.20/restconf/data/openconfig-interfaces:interfaces/
interface=Ethernet1/openconfig-interfaces-ext:mapped-vlans/mapped-vlan=100" -H "accept:
/" -H "Authorization: Basic YWRtaW46YWRtaW4="

```

## Example: Configure and delete double-tagged CVLAN-to-SVLAN translation using REST API

```
curl -X PATCH "https://100.94.113.29/restconf/data/openconfig-interfaces:interfaces/interface=Ethernet8/openconfig-interfaces-ext:mapped-vlans" -H "accept: */*" -H "Authorization: Basic YWRtaW46YWRtaW4xMjM=" -H "Content-Type: application/yang-data+json" -d "{\"openconfig-interfaces-ext:mapped-vlans\": {\"mapped-vlan\": [{\"vlan-id\":100,\"config\":{\"vlan-id\":100},\"match\":{\"match-double-tags\":{\"match-double-tag\": [{\"outer-vlan\":10,\"inner-vlan\":20,\"config\":{\"inner-vlan\":10,\"outer-vlan\":20,\"priority\":3}}, {\"outer-vlan\":30,\"inner-vlan\":40,\"config\":{\"inner-vlan\":30,\"outer-vlan\":40,\"priority\":5}}, {\"outer-vlan\":50,\"inner-vlan\":60,\"config\":{\"inner-vlan\":50,\"outer-vlan\":60}}]}}, {\"ingress-mapping\":{\"config\":{\"vlan-stack-action\":\"SWAP\"}}, \"egress-mapping\":{\"config\":{\"vlan-stack-action\":\"SWAP\"}}}]}}}"
```

```
curl -X DELETE "https://100.94.113.29/restconf/data/openconfig-interfaces:interfaces/interface=Ethernet8/openconfig-interfaces-ext:mapped-vlans/mapped-vlan=100/match/match-double-tags/match-double-tag=10,20" -H "accept: */*" -H "Authorization: Basic YWRtaW46YWRtaW4xMjM="
```

```
curl -X DELETE "https://100.94.114.20/restconf/data/openconfig-interfaces:interfaces/interface=Ethernet1/openconfig-interfaces-ext:mapped-vlans/mapped-vlan=100" -H "accept: */*" -H "Authorization: Basic YWRtaW46YWRtaW4="
```

## Layer 3 subinterfaces

On a Layer 3 physical interface or port channel, you can configure virtual VLAN subinterfaces to route IPv4/IPv6 traffic. Configure a L3 VLAN subinterface with a unique IP address and L3 routing protocol parameters. Each L3 subinterface provides a unique virtual L3 interface for the VLANs supported on the parent L3 physical interface.

Use L3 subinterfaces for VLAN trunking, IP routing, and routing L2 traffic between VLANs. L3 subinterfaces support various features and protocols, such as ARP, NAT, NDP, unnumbered IPv4, static and dynamic routes, ECMP, VXLAN, BGP EVPN, MLAG (L3 port channel only), OSPF, BFD, DHCP and DHCP relay, IPSLA sessions, IP helper, VRRP, 802.1Q Tunneling (QinQ), QoS (inherited from parent interface), multicast routing (IGMP and PIM-SSM), and SPAN/ERSPAN.

### Configure L3 subinterfaces

You can configure multiple L3 subinterfaces on a routed interface. The IP address of a subinterface must be in a different subnet from other subinterfaces on the L3 physical interface.

1. Configure the parent Ethernet or port channel interface for L3 routing:

```
sonic(config)# interface Eth1/4
sonic(conf-if-Eth1/4)# ip address ip-address/mask
sonic(conf-if-Eth1/4)# ipv6 address ipv6-address/prefix-length
sonic(conf-if-Eth1/4)# exit
```

2. Create a L3 subinterface on a physical Ethernet or port channel interface, and enter Subinterface Configuration mode, where *subinterface* is from 1 to 65535.

```
sonic(config)# interface Eth slot/port[/breakout-port].subinterface
sonic(conf-subif-Eth) #
```

```
sonic(config)# interface PortChannel port-channel-number.subinterface
sonic(conf-subif-PortChannel) #
```

For example:

```
sonic(config)# interface Eth1/4.10
sonic(conf-subif-Eth1/4.10)#
sonic(config)# interface PortChannel 100.100
sonic(conf-subif-PortChannel100.100) #
```

3. Configure a L3 VLAN on the subinterface. A L3 subinterface inherits its speed and MTU from the parent interface. The VLAN ID range is from 1 to 4094:

```
sonic(conf-subif-Eth1/4.10) # encapsulation dot1q vlan-id vlan-id
```

```
sonic(conf-PortChannel14.100) # encapsulation dot1q vlan-id vlan-id
```

To remove the VLAN from the subinterface, enter the no encapsulation dot1q *vlan-id* command.

4. Configure a unique IPv4/IPv6 address on the subinterface.

```
sonic(config) # interface Eth1/4.10
sonic(conf-subif-Eth1/4.10) # ip address 4.10.1.1/24
sonic(conf-subif-Eth1/4.10) # ipv6 address 410::1/64
```

5. Enable the L3 subinterface for incoming and outgoing traffic.

```
sonic(conf-subif-Eth1/4.10) # no shutdown
```

6. Verify L3 subinterface configuration and status by entering the show ip interfaces or show subinterfaces status command.

#### **Example: Configure L3 subinterfaces**

```
sonic(config) # interface Eth 1/26.100
sonic(conf-subif-Eth1/26.100) # encapsulation dot1q vlan-id 100
sonic(conf-subif-Eth1/26.100) # ip address 100.0.0.1/24
sonic(conf-subif-Eth1/26.100) # ipv6 address 100::1/64
sonic(conf-subif-Eth1/26.100) # exit

sonic(config) # interface PortChannel 3.101
sonic(conf-subif-PortChannel3.101) # encapsulation dot1q vlan-id 101
sonic(conf-subif-PortChannel3.101) # ip address 101.0.0.1/24
sonic(conf-subif-PortChannel3.101) # ipv6 address 101::1/64
sonic(conf-subif-PortChannel3.101) # exit
sonic(config) # exit

sonic# show subinterfaces status

Sub port interface Speed MTU Vlan Admin Type

Eth1/26.100 100000 9100 100 up dot1q-encapsulation
PortChannel3.101 25000 9100 101 up dot1q-encapsulation

sonic# show ip interfaces
Flags: U-Unnumbered interface, A-Anycast IP

Interface IP address/mask VRF Admin/Oper Flags

Vlan101 172.16.36.1/24 VrfTenant up/up A
Vlan102 172.16.38.1/24 VrfTenant up/up A
Vlan201 172.16.37.1/24 VrfTenant up/up A
Vlan50 172.16.39.1/24 VrfTenant up/up A
Eth1/26.100 100.0.0.1/24 VrfTenant up/up A
Loopback0 192.168.200.4/32 VrfTenant up/up A
Loopback1 192.168.10.2/32 VrfTenant up/up A
PortChannel3.101 101.0.0.1/24 VrfTenant up/up A
Vlan4094 192.168.100.5/31 VrfTenant up/down A
Management0 172.17.95.14/24 VrfTenant up/up A
```

## Show transceivers

To view the transceivers installed in Ethernet ports, use the show interface transceiver command.

- (i) NOTE:** Installed transceivers are automatically configured in switch ports. No user intervention is necessary. Transceiver auto-configuration is performed independently of autonegotiation, and works both when autonegotiation is enabled or disabled. Similarly, media-based port settings, such as link training and FEC, work independently of transceiver auto-configuration and are manually configured user settings.

## View installed transceivers

```
sonic# show interface transceiver

Eth1/1

Attribute : Value/State

cable-length(m) : 15
connector-type : OPTICAL-PIGTAIL
date-code : 2019-05-23
display-name : QSFP56-DD 400GBASE-SR8-AOC-15.0M
form-factor : QSFP56-DD
max-module-power(Watts) : 12
max-port-power(Watts) : 12
qualified : True
present : PRESENT
serial-no : CN04HQ0094GC016
vendor : DELL
vendor-oui : AC-4A-FE
vendor-part : JMCWK
vendor-rev : X1
...
Eth1/2

Attribute : Value/State

cable-length(m) : 15
connector-type : OPTICAL-PIGTAIL
date-code : 2019-05-23
display-name : QSFP56-DD 400GBASE-SR8-AOC-15.0M
form-factor : QSFP56-DD
max-module-power(Watts) : 12
max-port-power(Watts) : 12
qualified : True
present : PRESENT
serial-no : CN04HQ0094GC016
vendor : DELL
vendor-oui : AC-4A-FE
vendor-part : JMCWK
vendor-rev : X1
...
Eth1/33

Attribute : Value/State

cable-length(m) : 100
connector-type : RJ45
date-code : 2015-05-27
display-name : SFP 1000BASE-T
form-factor : SFP
max-module-power(Watts) : 1.5
max-port-power(Watts) : 2.5
qualified : True
present : PRESENT
serial-no : PTM3BJW
vendor : FINISAR CORP.
vendor-oui : 00-90-65
vendor-part : FCLF8522P2BTL
vendor-rev : A
Eth1/34

Attribute : Value/State

cable-length(m) : 30
connector-type : RJ45
date-code : 01
display-name : SFP+ 10GBASE-T
form-factor : SFP+
max-module-power(Watts) : 2
```

```

max-port-power(Watts) : 2.5
qualified : True
present : PRESENT
serial-no : 16430315
vendor : DELL
vendor-oui : 00-90-65
vendor-part : PGYJT
vendor-rev : A053

```

**i | NOTE:** To view the show output in one display without having to page through screen displays, enter `show interface transceiver | no-more`.

```
sonic# show interface transceiver summary
```

| Interface | Name                             | Vendor  | Part No.         | Serial No.      | Qualified |
|-----------|----------------------------------|---------|------------------|-----------------|-----------|
| Eth1/1    | QSFP56-DD 400GBASE-SR8-AOC-15.0M | DELL    | JMCWK            | CN04HQ0094GC016 | True      |
| Eth1/2    | QSFP56-DD 400GBASE-SR8-AOC-15.0M | DELL    | JMCWK            | CN04HQ0094GC016 | True      |
| Eth1/3    | QSFP56-DD 400GBASE-SR8-AOC-15.0M | DELL    | JMCWK            | CN04HQ0094GC019 | True      |
| Eth1/4    | QSFP56-DD 400GBASE-SR8-AOC-15.0M | DELL    | JMCWK            | CN04HQ0094GC019 | True      |
| Eth1/5    | QSFP56-DD 400GBASE-SR8-AOC-15.0M | Hisense | DMQ8811-EC10-DEN | UH293G00031     | True      |
| Eth1/6    | QSFP56-DD 400GBASE-SR8-AOC-15.0M | Hisense | DMQ8811-EC10-DEN | UH293G00032     | True      |
| Eth1/7    | SFP56-DD 400GBASE-SR8-AOC-15.0M  | Hisense | LMQ8811-PC+-DEN  | UBG93G00016     | True      |
| Eth1/8    | QSFP56-DD 400GBASE-SR8-AOC-15.0M | Hisense | LMQ8811-PC+-DEN  | UBG93G00010     | True      |
| Eth1/9    | QSFP56-DD 400GBASE-SR8-AOC-15.0M | Hisense | LMQ8811-PC+-DEN  | UBG93G00012     | True      |
| Eth1/10   | N/A                              | N/A     | N/A              | N/A             | N/A       |
| ...       |                                  |         |                  |                 |           |

```
sonic# show interface transceiver Eth1/2
```

```
Eth1/2
```

| Attribute               | : Value/State                      |
|-------------------------|------------------------------------|
| connector-type          | : OPTICAL-PIGTAIL                  |
| date-code               | : 2019-05-23                       |
| form-factor             | : QSFP56-DD                        |
| cable-length(m)         | : 15                               |
| display-name            | : QSFP56-DD 400GBASE-SR8-AOC-15.0M |
| max-module-power(Watts) | : 12                               |
| max-port-power(Watts)   | : 12                               |
| vendor-oui              | : AC-4A-FE                         |
| present                 | : PRESENT                          |
| serial-no               | : CN04HQ0094GC016                  |
| vendor                  | : DELL                             |
| vendor-part             | : JMCWK                            |
| vendor-rev              | : X1                               |

**i | NOTE:** In the `show interface transceiver` output, N/A or NOT-PRESENT indicate that no transceiver is installed.

```
sonic# show interface transceiver Eth1/9
```

```
Eth1/9
```

| Attribute | : Value/State |
|-----------|---------------|
| present   | : NOT-PRESENT |

**i** **NOTE:** If standard interface-naming mode is enabled, you must enter the Ethernet interface in the format show interface transceiver Eth slot/port[/breakout-port]; for example, show interface transceiver Eth 1/1/1.

### View DOM statistics

Use Digital Optical Monitoring (DOM) to monitor real-time operation of a transceiver in an Ethernet port.

```
sonic# show interface transceiver Ethsport dom
```

Using DOM, you can monitor the transmit and receive power of the transceiver, its temperature, and supply voltage. This information is useful to diagnose issues concerning an interface's link status; for example:

```
sonic# show interface transceiver Eth1/10 dom

Eth1/10

Identifier: SFP
 Vendor Name: DELL
 Vendor Part: 3P3PG
 ModuleMonitorValues:
 Temperature: 24.87 C
 Vcc: 3.30 Volts
 ChannelMonitorValues:
 Rx1Power: -1.70 dBm
 Tx1Bias: 6.76 mA
 Tx1Power: -1.61 dBm
 ChannelThresholdValues:
 RxPowerHighAlarm : 3.40 dBm
 RxPowerHighWarning: 2.40 dBm
 RxPowerLowAlarm : -10.50 dBm
 RxPowerLowWarning : -9.50 dBm
 TxBiasHighAlarm : 12.00 mA
 TxBiasHighWarning : 10.00 mA
 TxBiasLowAlarm : 0.00 mA
 TxBiasLowWarning : 0.00 mA
 TxPowerHighAlarm : 2.90 dBm
 TxPowerHighWarning: 2.40 dBm
 TxPowerLowAlarm : -8.60 dBm
 TxPowerLowWarning : -8.10 dBm
 ModuleThresholdValues:
 TempHighAlarm : 80.00 C
 TempHighWarning: 75.00 C
 TempLowAlarm : -10.00 C
 TempLowWarning : -5.00 C
 VccHighAlarm : 3.60 Volts
 VccHighWarning : 3.50 Volts
 VccLowAlarm : 3.00 Volts
 VccLowWarning : 3.10 Volts
```

**i** **NOTE:** In show transceiver dom summary output, only information about Rx/Tx power from the first channel on the transceiver EEPROM from all interfaces is displayed. To display complete information from all lanes on the transceiver EEPROM for an interface, use the show interface transceiver dom Ethsport command.

**i** **NOTE:** When a 400G port is broken out into 2x100G mode to support a single QSFP28 transceiver, the DOM information for the transceiver is displayed under both breakout interfaces; for example:

```
sonic# show interface breakout port 1/13

Port Breakout Mode Status Interfaces

1/13 2x100G Completed Eth1/13/1
 Eth1/13/2

sonic# show interface transceiver dom summary | grep 1/13
Eth1/13/1 QSFP28 LUMENTUM 22.30 3.27 -1.08 -0.22
Eth1/13/2 QSFP28 LUMENTUM 22.30 3.27 -1.33 -0.02
```

When you use a single QSFP28 transceiver in a 400G port, consider only the DOM information for the first 100G port of the 2x100G breakout.

## Media-based port autoconfiguration

When enabled, media-based port autoconfiguration allows a port's configuration settings to be automatically updated according to the installed transceiver.

Transceivers are automatically configured when installed in switch ports. No user intervention is necessary. Transceiver auto-configuration is performed independently of autonegotiation, and works both when autonegotiation is enabled or disabled. However, media-based port settings, such as link training and FEC, work independently of transceiver auto-configuration and are manually configured settings.

By default, an installed transceiver is enabled to automatically configure some features and is disabled for others. For example, [Forward Error Correction](#) (FEC) is not automatically configured on a port by the installed transceiver. You must manually set the FEC type. However, [Loss of Signal](#) (LOS) is automatically enabled on a port by media-based autoconfiguration, and works without user intervention.

## High-power optics

You can enable or disable high-power optics on a port.

PowerSwitch Z9332F-ON and Z9432F-ON devices running Enterprise SONiC use higher-power optics that utilize a significant percentage of the overall power budget at the risk of causing an entire power bank to shut down, causing service disruption. The system also displays appropriate warning logs when high-power optic is inserted into a port. For more information about high-power optics, see *Enterprise SONiC Distribution high-power optics*.

By default, high-power optics feature is enabled on all the physical interfaces. SONiC enables or disables high-power optics based on the warning threshold and alarm threshold.

- If you have enabled high-power optics on a port and if maximum media power exceeds the maximum port alarm power, the system disables the optic.
- If you have disabled high-power optics on a port and if the maximum media power exceeds the maximum port warn power, the system disables the optic.

### Enable high-power optics on a port

You can enable or disable high-power optics on a port.

- To disable high-power optics on a port, enter the no version of the command in INTERFACE mode.

```
sonic(conf-if-Eth1/1)# no allow high-wattage-optics
```

- To enable an optic that was disabled earlier, enter the allow high-wattage-optics command in INTERFACE mode.

```
sonic(conf-if-Eth1/1)# allow high-wattage-optics
```

The system displays appropriate syslog message when a high-power optic is disabled or enabled. For example:

```
Jul 07 11:25:39.986906+00:00 2023
MAA-Z9432F-ON-A03-11862 NOTICE pmon#xcvrd[80]: Media in port Eth1/1 serial number
L2125D0382 is high-wattage-optic and is disabled
```

```
Jul 04 12:04:43.350675+00:00 2023 sonic NOTICE pmon#xcvrd[88]:
Media in port Ethernet0 serial number L2125D0382 is high-wattage-optic
```

To view the information about the high-power optics in the system:

```
sonic# show interface transceiver wattage

Interface Media state Media-Max-Power Port-Max-Power High-Power-Media Media-Lockdown-
```

| -----   |                                 |     |    |       |       |  |
|---------|---------------------------------|-----|----|-------|-------|--|
| Eth1/1  | QSFP56-DD 400GBASE-ZR           | 18  | 15 | True  | True  |  |
| Eth1/2  | Not Present                     |     |    |       |       |  |
| Eth1/3  | QSFP56-DD 400GBASE-CR8-DAC-0.5M | 1.5 | 15 | False | False |  |
| Eth1/4  | Not Present                     |     |    |       |       |  |
| Eth1/5  | QSFP56-DD 400GBASE-CR8-DAC-0.5M | 1.5 | 15 | False | False |  |
| Eth1/6  | Not Present                     |     |    |       |       |  |
| Eth1/7  | QSFP+ 40GBASE-CR4-DAC-1.0M      | 1.5 | 15 | False | False |  |
| Eth1/8  | QSFP28 100GBASE-CR4-DAC-1.0M    | 1.5 | 15 | False | False |  |
| Eth1/9  | QSFP56-DD 400GBASE-ZR           | 19  | 20 | True  | False |  |
| Eth1/10 | QSFP+ 40GBASE-SR4               | 1.5 | 15 | False | False |  |
| Eth1/11 | Not Present                     |     |    |       |       |  |
| Eth1/12 | Not Present                     |     |    |       |       |  |
| Eth1/13 | Not Present                     |     |    |       |       |  |
| Eth1/14 | Not Present                     |     |    |       |       |  |
| Eth1/15 | QSFP28 100GBASE-CR4-DAC-1.0M    | 1.5 | 15 | False | False |  |
| Eth1/16 | Not Present                     |     |    |       |       |  |
| Eth1/17 | Not Present                     |     |    |       |       |  |
| Eth1/18 | Not Present                     |     |    |       |       |  |
| Eth1/19 | Not Present                     |     |    |       |       |  |
| Eth1/20 | Not Present                     |     |    |       |       |  |
| Eth1/21 | Not Present                     |     |    |       |       |  |
| Eth1/22 | Not Present                     |     |    |       |       |  |
| Eth1/23 | QSFP+ 40GBASE-CR4-DAC-1.0M      | 1.5 | 20 | False | False |  |
| Eth1/24 | Not Present                     |     |    |       |       |  |
| Eth1/25 | Not Present                     |     |    |       |       |  |
| Eth1/26 | Not Present                     |     |    |       |       |  |
| Eth1/27 | Not Present                     |     |    |       |       |  |
| Eth1/28 | QSFP+ 40GBASE-CR4-DAC-2.0M      | 1.5 | 15 | False | False |  |
| Eth1/29 | Not Present                     |     |    |       |       |  |
| Eth1/30 | Not Present                     |     |    |       |       |  |
| Eth1/31 | Not Present                     |     |    |       |       |  |
| Eth1/32 | Not Present                     |     |    |       |       |  |
| Eth1/33 | Not Present                     |     |    |       |       |  |
| Eth1/34 | Not Present                     |     |    |       |       |  |

# Layer 2

|                                                 |                                                                                                                                                                                                                        |
|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Link aggregation control protocol (LACP)</b> | Exchanges information between two systems and automatically establishes a link aggregation group (LAG) between systems (see <a href="#">Link aggregation control protocol</a> ).                                       |
| <b>Link layer discovery protocol (LLDP)</b>     | Enables a local area network (LAN) device to advertise its configuration and receive configuration information from adjacent LLDP-enabled infrastructure devices (see <a href="#">Link layer discovery protocol</a> ). |
| <b>Media access control (MAC)</b>               | Configures limits, redundancy, balancing, and failure detection settings for devices on your network (see <a href="#">Media access control</a> ).                                                                      |
| <b>Spanning-tree protocol (STP)</b>             | Prevents Layer 2 loops in a network and provides redundant links (see <a href="#">Spanning-tree protocol</a> ).                                                                                                        |
| <b>Port monitoring</b>                          | Monitors ingress and/or egress traffic of one port to another for analysis (see <a href="#">Port monitoring</a> ).                                                                                                     |

## Topics:

- [Link aggregation control protocol](#)
- [Link layer discovery protocol](#)
- [Media access control](#)
- [Spanning-tree protocol](#)
- [Port monitoring](#)
- [Port Security](#)

## Link aggregation control protocol

The link aggregation control protocol (LACP) dynamically bundles multiple physical ports in a logical port channel or link aggregation group (LAG). A port channel is automatically created through the exchange of LACP packets between ports.

All port channel member ports be the same speed and configured as either Layer 2 or Layer 3 interfaces. Traffic is load-balanced across all bundled links. If one member link fails, the LAG port channel continues to carry traffic over the remaining links.

### LACP modes

LACP operates on an interface in Active and On modes.

|               |                                                                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Active</b> | LACP runs on any interface/link that is configured in this mode. A port in Active mode automatically initiates negotiations with other ports using LACP packets, and sets up a port channel with another port that is in Active mode. |
| <b>On</b>     | The interface acts as a member of a static LAG. Only manually assigned interfaces are included as members of a port channel in On mode.                                                                                               |

When you add an interface as a member of a port channel, the interface is assigned the LACP mode that is configured for the port channel (see [Port channel configuration](#)). For example, if you configure the port channel to operate in On mode, all interfaces assigned to it operate in On mode.

```
sonic# interface portchannel 10 mode on
sonic(conf-if-po10)# exit
sonic(config)# interface Eth1/4
sonic(conf-if-Eth1/4)# channel-group 10
```

### View LACP-enabled interfaces

To display the LACP-enabled port channels and their member interfaces, use the `show interface PortChannel` command. In the show output, mode LACP indicates the interfaces that are dynamically bundled by LACP in port channels.

```
sonic# show interface PortChannel
PortChannel1 is up, line protocol is down, mode LACP
Minimum number of links to bring PortChannel up is 1
Fallback: Enabled
MTU 9100
LACP mode ACTIVE interval SLOW priority 65535 address ec:f4:bb:fc:09:8b
Members in this channel: Eth1/2
LACP Actor port 65 address ec:f4:bb:fc:09:8b key 1
LACP Partner port 0 address 00:00:00:00:00:00 key 0
Members in this channel: Eth1/3
LACP Actor port 73 address ec:f4:bb:fc:09:8b key 1
LACP Partner port 0 address 00:00:00:00:00:00 key 0
Last clearing of "show interface" counters: 1970-01-01 00:00:00
Input statistics:
 0 packets, 0 octets
0 Multicasts, 0 Broadcasts, 0 Unicasts
0 error, 0 discarded
Output statistics:
 0 packets, 0 octets
0 Multicasts, 0 Broadcasts, 0 Unicasts
0 error, 0 discarded
```

## Link layer discovery protocol

The link layer discovery protocol (LLDP) enables devices on a local area network (LAN) to advertise and learn the capabilities of adjacent LAN devices. LLDP transmits and receives information from LLDP agents on other LAN devices. By default, LLDP is enabled on all interfaces.

- An LLDP-enabled interface supports only one neighbor.
- The switch receives and periodically transmits LLDP protocol data units (PDUs). The default transmission interval is 30 seconds.
- LLDP PDU information received from a neighbor expires after a specific amount of time, called time to live (TTL). The default TTL value is 120 seconds.
- Spanning-tree blocked ports allow LLDP PDUs.
- Link layer discovery protocol-media endpoint discovery (LLDP-MED) is enabled on all interfaces by default.

### View LLDP neighbors

```
sonic# show lldp table

LocalPort RemoteDevice RemotePortID Capability RemotePortDescr

Ethernet80 AG2 Ethernet12 R Eth1/4
Ethernet84 AG2 Ethernet16 R Eth1/5
Ethernet88 AG2 Ethernet20 R Eth1/6
Ethernet92 AG1 Ethernet0 R Eth1/1/1
Ethernet96 AG1 Ethernet8 R Eth1/2/1
Ethernet100 AG1 Ethernet16 R Eth1/3/1
```

```
sonic# show lldp neighbor
Interface: Ethernet48,via: LLDP
Chassis:
 ChassisID: 3c:2c:30:6d:72:80
 SysName: 10025
 SysDescr: SONiC Software
 TTL: 120
 MgmtIP: 100.104.78.100
 MgmtIP: fe80::3e2c:30ff:fe6d:7280
 Capability: ROUTER, ON
Port
 PortID: Ethernet48
 PortDescr: Eth1/13
 PortVlanID: 10
```

```

LLDP-MED
Device Type: Network Connectivity Device
Capability: Capabilities, yes
Capability: Ext_mdi_power_pd, yes
Capability: Inventory, yes
Capability: Network_policy, yes
Inventory
Hardware Rev: 3.40.0.9-10
Software Rev: 5.10.0-8-2-amd64
Firmware Rev: 3.40.0.9-10
Serial Number: CN01WJVTCES0094Q0015
Manufacturer: Dell EMC
Model: S5232F-ON

```

## TLVs in LLDP

LLDP protocol data units (LLDPDUs) are sent in Ethernet frames from LLDP-enabled ports to neighbor devices to advertise switch configuration and capabilities. LLDPDUs carry different categories of type, length, and value (TLV) information, such as basic management TLVs, organizationally specific TLVs, LLDP-MED TLVs, and so on. This section explains how to enable various TLVs.

### Configure the system to send TLVs

You can configure the system to send certain TLVs in LLDPDUs. To enable the TLVs, follow these steps:

- Use the following command in CONFIGURATION or INTERFACE mode:

```
lldp tlv-select {management-address | system-capabilities | power-management | port-vlan-id}
```

- **management-address** — Send the management IP address in the TLV. Configure from CONFIGURATION mode.
- **system-capabilities** — Send system capabilities in the TLV. Configure from CONFIGURATION mode.
- **power-management** — Send power management information in the TLV. Configure from INTERFACE mode.
- **port-vlan-id** — Send the port's VLAN ID that is associated with untagged frames in the TLV. Configure from INTERFACE mode. By default, this option is enabled.

### Example

The following example enables the advertisement of VLAN ID of the port in the TLV:

```

sonic(config)# interface Ethernet 48
sonic(config-if-Ethernet48)# switchport access Vlan 10
sonic(config-if-Ethernet48)# lldp tlv-select port-vlan-id

```

### View TLVs received from a peer device

Use the following command to view the TLVs that are received from a peer device:

```

sonic# show lldp neighbor

Interface: Ethernet48,via: LLDP
Chassis:
 ChassisID: 3c:2c:30:6d:72:80
 SysName: 10025
 SysDescr: SONiC Software
 TTL: 120
 MgmtIP: 100.104.78.100
 MgmtIP: fe80::3e2c:30ff:fe6d:7280
 Capability: ROUTER, ON
Port
 PortID: Ethernet48
 PortDescr: Eth1/13
 PortVlanID: 10
LLDP-MED
 Device Type: Network Connectivity Device
 Capability: Capabilities, yes
 Capability: Ext_mdi_power_pd, yes
 Capability: Inventory, yes
 Capability: Network_policy, yes
Inventory
 Hardware Rev: 3.40.0.9-10
 Software Rev: 5.10.0-8-2-amd64
 Firmware Rev: 3.40.0.9-10
 Serial Number: CN01WJVTCES0094Q0015

```

Manufacturer: Dell EMC  
Model: S5232F-ON

## Media access control

A media access control (MAC) address is a 48-bit number in the format *nn.nn.nn.nn.nn.nn*. The MAC address table contains static and dynamic MAC address entries. Static MAC addresses are user-configured entries which do not age out. Dynamically-learned MAC addresses are hardware-based entries which age out according to the configured aging time.

When the switch receives a packet, it learns the MAC address in the source MAC address field on the port that the packet is received. The switch then looks up the destination MAC address for the specified VLAN in the MAC address table. If the destination MAC address is found, the switch forwards the packet to the appropriate port/port channel. If the destination MAC address is not found, the switch floods the packet on all VLAN ports.

### Configure a static MAC address

To add a static MAC address to the MAC address table, manually configure the address. Specify the Ethernet port or port channel and VLAN through which the device with the static MAC address can be reached and to which the switch can forward packets.

```
sonic(config)# mac address-table mac-address Vlan vlan-id {Eth slot/port[/breakout-port]
|
PortChannel number}
```

To delete a static MAC address, use the no version of the complete command without the Ethernet port or port channel number; for example:

```
sonic(config)# no mac address-table 00:00:00:00:00:01 Vlan 10
```

### Configure aging time

Configure the aging time for all dynamically-learned MAC addresses (0 to 1000000 seconds; default 600). Static MAC address entries are not affected by the mac address-table aging-time command. When the aging time is reached, a dynamic MAC address entry is deleted from the table. Enter 0 to disable MAC aging. Enter the no version of the command to restore the default aging time.

```
sonic(config)# mac address-table aging-time seconds
```

For example:

```
sonic# show mac address-table aging-time
Global aging time: 600 seconds

sonic(config)# mac address-table aging-time 1000
sonic# show mac address-table aging-time
Global aging time: 1000 seconds

sonic(config)# mac address-table aging-time 0
sonic# show mac address-table aging-time
Global aging time: 0 seconds (disabled)

sonic(config)# no mac address-table aging-time
sonic(config)# exit
sonic# show mac address-table aging-time
Global aging time: 600 seconds
```

### Configure MAC address dampening

To limit the maximum number times that a dynamic MAC address can be assigned to different interfaces, configure a dampening threshold (5 to 100; no default).

```
sonic(config)# mac address-table dampening-threshold number
```

To configure the minimum time interval that a dynamic MAC address can be assigned to different interfaces, configure a dampening interval (1 to 1000000 seconds; no default).

```
sonic(config)# mac address-table dampening-interval seconds
```

### Static MAC address configuration

```
sonic(config)# mac address-table 00:00:00:00:00:01 Vlan 10 Eth1/1
sonic(config)# mac address-table 00:00:00:00:00:01 Vlan 11 Eth1/1
sonic(config)# mac address-table 00:00:00:00:00:03 Vlan 30 Eth1/2
sonic(config)# mac address-table 00:00:00:00:00:05 Vlan 50 Eth1/3
sonic(config)# mac address-table 00:00:00:00:00:07 Vlan 70 Eth1/4
sonic(config)# mac address-table 00:00:00:00:00:09 Vlan 90 Eth1/5
sonic(config)# mac address-table 00:00:00:00:00:98 Vlan 10 Eth1/6
sonic(config)# mac address-table 00:00:00:00:00:99 Vlan 99 PortChannel 110
sonic(config)# mac address-table aging-time 1000
sonic(config)# mac address-table dampening-threshold 10
sonic(config)# mac address-table dampening-interval 100000
```

### View MAC address table

Use `show mac-address table` commands to view MAC address table information.

```
show mac address-table [address mac-address] [aging-time] [interface Eth slot/port[/breakout-port]] [Vlan vlan-id] [PortChannel number] [count] [static] [dynamic]
```

```
sonic# show mac address-table
```

| VLAN | MAC-ADDRESS       | TYPE    | INTERFACE      |
|------|-------------------|---------|----------------|
| 10   | 00:00:00:00:00:01 | STATIC  | Eth1/1         |
| 11   | 00:00:00:00:00:01 | STATIC  | Eth1/1         |
| 100  | 00:00:00:00:00:10 | DYNAMIC | Eth1/9         |
| 20   | 00:00:00:00:00:02 | DYNAMIC | Eth1/2         |
| 30   | 00:00:00:00:00:03 | STATIC  | Eth1/2         |
| 40   | 00:00:00:00:00:04 | DYNAMIC | Eth1/3         |
| 50   | 00:00:00:00:00:05 | STATIC  | Eth1/3         |
| 60   | 00:00:00:00:00:06 | DYNAMIC | Eth1/4         |
| 70   | 00:00:00:00:00:07 | STATIC  | Eth1/4         |
| 80   | 00:00:00:00:00:08 | DYNAMIC | Eth1/4         |
| 90   | 00:00:00:00:00:09 | STATIC  | Eth1/5         |
| 99   | 00:00:00:00:00:99 | STATIC  | PortChannel110 |

### View MAC address table entries by MAC address

```
sonic# show mac address-table address 00:00:00:00:00:01
```

| VLAN | MAC-ADDRESS       | TYPE   | INTERFACE |
|------|-------------------|--------|-----------|
| 10   | 00:00:00:00:00:01 | STATIC | Eth1/1    |
| 11   | 00:00:00:00:00:01 | STATIC | Eth1/1    |

### View MAC address table entries by VLAN

```
sonic# show mac address-table Vlan 10
```

| VLAN | MAC-ADDRESS       | TYPE   | INTERFACE |
|------|-------------------|--------|-----------|
| 10   | 00:00:00:00:00:01 | STATIC | Eth1/1    |
| 10   | 00:00:00:00:00:98 | STATIC | Eth1/1    |

```
sonic# show mac address-table static Vlan 11
```

| VLAN | MAC-ADDRESS | TYPE | INTERFACE |
|------|-------------|------|-----------|
|------|-------------|------|-----------|

```

11 00:00:00:00:00:01 STATIC Eth1/1
```

```
sonic# show mac address-table dynamic Vlan 60
```

```

VLAN MAC-ADDRESS TYPE INTERFACE

60 00:00:00:00:00:06 DYNAMIC Eth1/12
```

#### **View MAC address count**

```
sonic# show mac address-table count
MAC Entries for all vlans : 13
Dynamic Address Count : 5
Static Address (User-defined) Count : 8
Total MAC Addresses in Use: 13
```

#### **View MAC address entries by interface**

```
sonic# show mac address-table interface Eth1/1
```

```

VLAN MAC-ADDRESS TYPE INTERFACE

10 00:00:00:00:00:01 STATIC Eth1/1
10 00:00:00:00:00:98 STATIC Eth1/1
11 00:00:00:00:00:01 STATIC Eth1/1
```

```
sonic# show mac address-table static interface Eth1/3
```

```

VLAN MAC-ADDRESS TYPE INTERFACE

30 00:00:00:00:00:03 STATIC Eth1/3
```

```
sonic# show mac address-table dynamic interface Eth1/5
```

```

VLAN MAC-ADDRESS TYPE INTERFACE

60 00:00:00:00:00:06 DYNAMIC Eth1/5
```

#### **View MAC address entries by port channel**

```
sonic# show mac address-table interface PortChannel 10
```

```

VLAN MAC-ADDRESS TYPE INTERFACE

99 00:00:00:00:00:99 STATIC PortChannel10
```

```
sonic# show mac address-table static interface PortChannel 10
```

```

VLAN MAC-ADDRESS TYPE INTERFACE

99 00:00:00:00:00:99 STATIC PortChannel10
```

#### **View static MAC address entries**

```
sonic# show mac address-table static
```

```

VLAN MAC-ADDRESS TYPE INTERFACE

10 00:00:00:00:00:01 STATIC Eth1/1
11 00:00:00:00:00:01 STATIC Eth1/1
30 00:00:00:00:00:03 STATIC Eth1/3
50 00:00:00:00:00:05 STATIC Eth1/5
70 00:00:00:00:00:07 STATIC Eth1/7
90 00:00:00:00:00:09 STATIC Eth1/9
```

```

10 00:00:00:00:00:98 STATIC Eth1/1
99 00:00:00:00:00:99 STATIC PortChannel10

```

```

sonic# show mac address-table static address 00:00:00:00:00:01

VLAN MAC-ADDRESS TYPE INTERFACE

10 00:00:00:00:00:01 STATIC Eth1/1
11 00:00:00:00:00:01 STATIC Eth1/1

```

### View dynamic MAC address entries

```

sonic# show mac address-table dynamic

VLAN MAC-ADDRESS TYPE INTERFACE

100 00:00:00:00:00:10 DYNAMIC Eth1/1
20 00:00:00:00:00:02 DYNAMIC Eth1/2
40 00:00:00:00:00:04 DYNAMIC Eth1/3
60 00:00:00:00:00:06 DYNAMIC Eth1/4
80 00:00:00:00:00:08 DYNAMIC Eth1/5

```

```

sonic# show mac address-table dynamic address 00:00:00:00:00:06

VLAN MAC-ADDRESS TYPE INTERFACE

60 00:00:00:00:00:06 DYNAMIC Eth1/1

```

## Spanning-tree protocol

**i** **NOTE:** Multiple Spanning Tree Protocol (MSTP), Per-VLAN Spanning-Tree plus (PVST+), and Rapid-PVST plus (RPVST+) are supported only in the Enterprise Standard, Enterprise Premium, and Edge Standard bundles. MSTP, PVST+, and RPVST+ are not supported in the Cloud Standard and Cloud Premium bundles.

Spanning-tree protocol (STP) prevents Layer 2 loops in a network and provides redundant links. If a primary link fails, the backup link is activated and network traffic is not affected. STP also ensures that the least cost path is taken when multiple paths exist between the devices.

When spanning tree is used, the network switches transform the real network topology into a spanning tree topology. In an STP topology, any LAN in the network can be reached from any other LAN through a unique path. The network switches recalculate a new spanning tree topology whenever there is a change to the network topology.

For each switch in a topology, a port with the lowest path cost to the root bridge is elected as root port. For each LAN, the switches that attach to the LAN select a designated switch that is the closest to the root switch. The designated switch forwards all traffic to and from the LAN.

The port on the designated switch that connects to the LAN is called the *designated port*. The switches decide which of their ports are part of the spanning-tree. A port is in the spanning-tree if it is a root port or a designated port.

PVST+ allows for running multiple instances of spanning-tree on a per-VLAN basis. One of the advantages of PVST is it allows for load balancing of traffic. When a single instance of spanning-tree is run, and a link is put into blocking state for avoiding the loop, it results in inefficient bandwidth usage.

With per-VLAN spanning-tree, multiple instances can be run such that for some of the instances traffic is blocked over the link, and for other instances traffic is forwarded allowing for traffic load balancing.

PVST+ support allows the device to interoperate with IEEE STP and also tunnel the PVST+ BPDUs transparently across IEEE STP region to potentially connect other PVST+ switches across the IEEE STP region. For interop with IEEE STP, PVST+ will send untagged IEEE BPDUs (MAC - 01:80:C2:00:00:00) with information corresponding to VLAN 1. The STP port must be a member of VLAN 1 for interoperating with IEEE STP.

## About STP

STP is a Layer 2 link management protocol that provides path redundancy while preventing undesirable loops in the network. For a Layer 2 network to function properly, only one active path can exist between any two stations.

A loop-free subset of a network topology is called a spanning-tree. The operation of a spanning-tree is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

A Dell PowerSwitch series switch uses STP on all VLANs. By default, a single spanning-tree runs on each configured VLAN (provided you do not manually disable the spanning-tree). You can enable and disable a spanning-tree on a per-VLAN basis.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The spanning-tree algorithm calculates the best loop-free path throughout a switched Layer 2 network. Switches send and receive spanning-tree frames at regular intervals. The switches do not forward these frames, but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and switches might learn end station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network.

A spanning-tree defines a tree with a root switch and a loop-free path from the root to all switches in the Layer 2 network. A spanning-tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning-tree fails and a redundant path exists, the spanning-tree algorithm recalculates the spanning-tree topology and activates the standby path.

When two ports on a switch are part of a loop, the spanning-tree port priority and port path cost setting determine which port is put in the forwarding state and which port is put in the blocking state. The spanning-tree port priority value represents the location of an interface in the network topology and how well located it is to pass traffic. The spanning-tree port path cost value represents media speed.

## Change STP mode

This information describes how to configure STP on port-based VLANs. The switch can use the per-VLAN spanning-tree plus (PVST+) protocol based on the IEEE 802.1D standard, or the rapid per-VLAN spanning-tree plus (rapid-PVST+) protocol based on the IEEE 802.1w standard.

The default xSTP variant running in Enterprise SONiC is Rapid-PVST. You can change the mode with the `spanning-tree mode {pvst | rapid-pvst | mst}` command.

## Enable or disable STP

You can disable the STP globally on the switch or at the interface level. Disabling spanning tree at an instance level causes all the port members of that instance to disable the spanning tree. This moves the port to the Forwarding/Blocking state based on the operational status of the ports.

Use `spanning-tree enable` to enable STP on a per-port basis, or use `no spanning-tree enable` to disable STP.

## Enable BPDU filtering

Follow these steps to enable BPDU filtering on an interface.

1. Configure the interface for BPDU filtering.

```
sonic(config)# interface phy-if-name number
```

2. Enable spanning-tree BPDU guard to shut down an interface when it receives a BPDU.

```
sonic(conf-if-Eth1/2)# spanning-tree bpduguard port-shutdown
```

3. Configure the spanning-tree cost (0 to 65535) on an interface.

```
sonic(conf-if-Eth1/2)# spanning-tree cost value
```

4. Enable spanning-tree on an interface.

```
sonic(conf-if-Eth1/2)# spanning-tree enable
```

5. Enable spanning-tree port fast mode on an interface.

```
sonic(conf-if-Eth1/2) # spanning-tree portfast
```

6. Configure the port level priority value (0 to 255 in increments of 4; default 128).

```
sonic(conf-if-Eth1/2) # spanning-tree port-priority value
```

7. Configure spanning-tree uplink fast on an interface.

```
sonic(conf-if-Eth1/2) # spanning-tree uplinkfast
```

8. Enable BPDU filtering.

```
sonic(conf-if-Eth1/2) # spanning-tree bpdufilter enable
```

9. Enable loop guard globally on all ports.

```
sonic(config) # spanning-tree loopguard default
```

10. Configure the spanning-tree BPDU filter globally on the switch in Configuration mode.

```
sonic(config) # spanning-tree edge-port bpdufilter default
```

### Verify configuration

```
sonic(conf-if-Eth1/2) # do show running-configuration spanning-tree
spanning-tree mode pvst
spanning-tree edge-port bpdufilter default
spanning-tree forward-time 4
spanning-tree guard root timeout 5
spanning-tree hello-time 1
spanning-tree max-age 40
spanning-tree priority 61440
!
no spanning-tree vlan 101
!
spanning-tree vlan 100 forward-time 5
spanning-tree vlan 100 hello-time 3
!
spanning-tree vlan 101 forward-time 15
!
interface Eth1/1
spanning-tree bpdufilter enable
spanning-tree vlan 100 cost 1
!
interface PortChannel10
no spanning-tree enable
no spanning-tree portfast
spanning-tree cost 2
spanning-tree vlan 100 cost 1
```

## Recover from BPDU guard violations

When there is BPDU guard violation on a port, Enterprise SONiC either shuts down the port or moves it to Blocked state.

Follow these steps to set the recovery timer value and recover the ports due to a BPDU guard violation.

1. Enable errdisable recovery.

```
sonic(config) # errdisable recovery cause bpduguard
```

When the recovery option is enabled, the port is brought up after the recovery timer expires. The default recovery timer value is 300 seconds. When the recovery option is disabled, the port remains shut down indefinitely. You must manually bring up the port with shutdown and no shutdown.

2. Change the recovery timer value (30 to 65535 seconds; default 300).

```
sonic(config) # errdisable recovery interval seconds
```

The interval value is applied only for shutdown.

#### Verify configuration

```
sonic# show errdisable recovery

Errdisable Cause Status
----- -----
udld disabled
bpduguard enabled
link-flap disabled
coa disabled
Timeout for Auto-recovery: 300 seconds
```

## Spanning-tree link type for rapid state transitions

Enterprise SONiC assumes a port that runs in full-duplex mode is a point-to-point link. A point-to-point link transitions to forwarding state faster. By default, Enterprise SONiC derives the link type of a port from the duplex mode.

By default, the port is set to point-to-point. If you want the port to be shared, the convergence will be slower. If you configure a port as a shared link, you cannot use the fast transition feature, regardless of the duplex setting.

- Configure the STP link type: point-to-point (full-duplex link) that accelerates STP state transitions or shared (half-duplex link).

```
sonic(conf-if-Ethslot/port)# spanning-tree link-type {point-to-point | shared}
```

## Dynamic path cost calculation

Path cost of an interface (physical or port channel) is calculated based on the speed of the port or port channel. When the speed of the port or port channel changes, the path cost recalculation is triggered based on the user-defined configuration.

You can enable/disable dynamic recalculation of path cost using the `spanning-tree path-cost` command. This command allows the protocol to do dynamic cost calculation whenever the channel-members are added or deleted. By default, this dynamic path cost calculation is enabled.

When dynamic path cost is disabled, protocol calculates the path cost when the port channel is coming up for the first time after creation or whenever dynamic path cost calculation is enabled and then disabled by management or when the user adds/removes member port to/from the port channel.

This feature allows the user to disable path cost recalculation on link flap events. If disabled, the path cost of the lag is calculated based on the below formula  $\text{LAG speed} = \text{speed of a single member} * \text{number of configured member ports}$  (irrespective of its operational status).

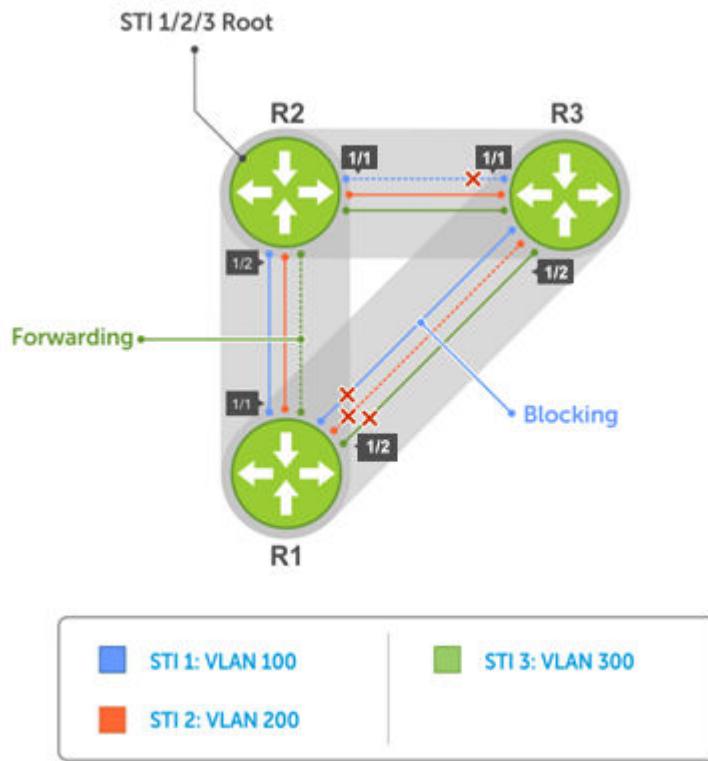
Path cost changes only for the user event [addition/removal of channel-member]. Path cost is calculated based on the number of configured ports. Dynamic path cost disable functionality is supported for VLT port channel.

## Rapid per-VLAN spanning-tree

Rapid per-VLAN spanning-tree (Rapid-PVST) is used to create a single topology per VLAN. Rapid-PVST is enabled by default; it provides faster convergence than STP and runs on the default VLAN (VLAN1).

Configuring Rapid-PVST is a four-step process:

1. Ensure that the interfaces are in L2 mode.
2. Place the interfaces in VLANs. By default, switchport interfaces are members of the default (VLAN1).
3. Enable Rapid-PVST. This step is only required if another variation of STP is present.
4. (Optional) Select a nondefault bridge-priority for the VLAN for load balancing.

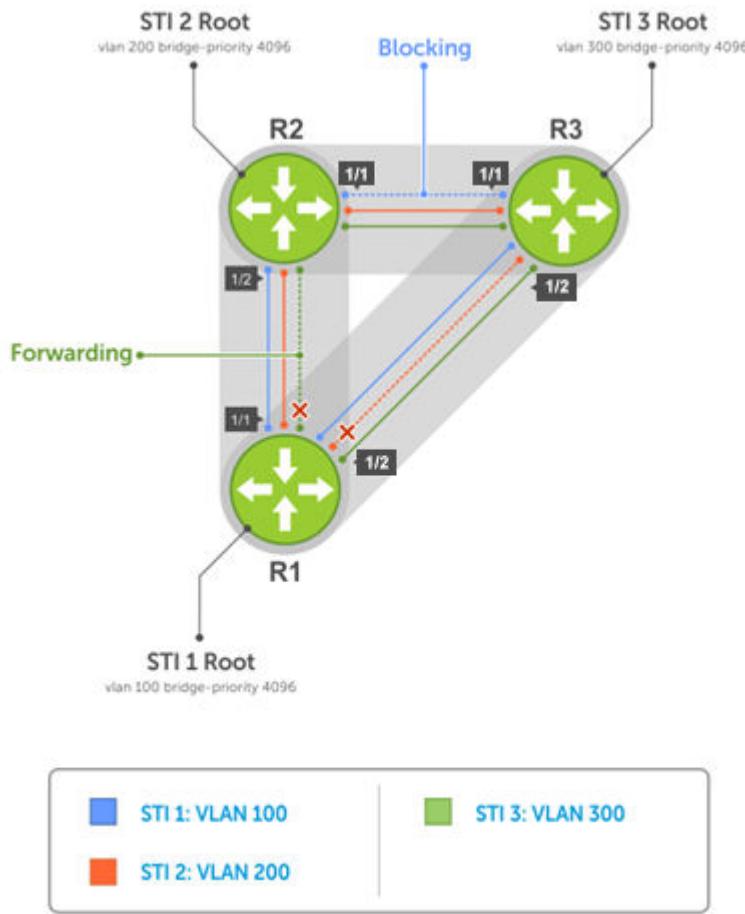


Each VLAN is assigned an incremental default bridge priority. For example, if VLAN 1 is assigned a bridge priority value of 32769, then VLAN 2 (if created) is assigned a bridge priority value of 32770. Similarly, VLAN 10 (if created) is assigned a bridge priority value of 32778, and so on. All three instances have the same forwarding topology.

**NOTE:** Z9332F-ON supports a total of 64 instances, of which three VLANs are used for internal purposes. When you run Rapid-PVST flavor, each VLAN allocates one instance until the VLAN count reaches 61 and map the default instance after that.

## Load balance root selection

By default, all VLANs use the same forwarding topology — R2 is elected as the root and all 10G Ethernet ports have the same cost. Bridge priority can be modified for each VLAN to enable different forwarding topologies.



To achieve Rapid-PVST load balancing, assign a different priority on each bridge.

## Enable Rapid-PVST

By default, Rapid-PVST is enabled and creates an instance during VLAN creation. To participate in Rapid-PVST, a port channel or physical interface must be a member of a VLAN. If another STP mode is enabled on the switch, re-enable Rapid-PVST.

- Enable Rapid-PVST mode.

```
sonic(config) # spanning-tree mode rapid-pvst
```

## Select root bridge

Rapid-PVST determines the root bridge according to VLAN bridge priority. Assign a bridge priority value to VLAN port members using the `spanning-tree vlan port-priority` command. The port members of VLANs with the lowest priority value are elected as a root bridge.

- Assign a bridge priority to VLAN port members (0 to 61440) or designate VLAN members as the root (0).

```
sonic(conf-if-Eth) # spanning-tree {vlan vlan-range port-priority priority-value}
```

- `vlan-range` — VLAN ID (1 to 4093)

- **priority value** — Priority value in increments of 4096 (default is 32768). The lower the number assigned, the more likely this bridge becomes the root bridge. The bridge priority valid values are: 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, or 61440. All other values are rejected.

To verify the Rapid-PVST root bridge configuration, use the `show running-configuration spanning-tree` command.

## Root assignment

Rapid-PVST assigns the root bridge according to the lowest VLAN bridge ID. Primary configuration assigns 24576 as the bridge priority, whereas secondary configuration assigns 28672 as the bridge priority.

The `spanning-tree vlan vlan-id root primary` command ensures that the switch has the lowest bridge priority value by setting the predefined value of 24576. If an alternate root bridge is required, use the `spanning-tree vlan vlan-id root secondary` command. The command sets the priority for the switch to the predefined value of 28672. If the primary root bridge fails, the command ensures that the alternate switch becomes the root bridge. It also assumes that the other switches in the network have a defined default priority value of 32768.

- Configure the device as the root or secondary root in CONFIGURATION mode.

```
spanning-tree vlan vlan-id root {primary | secondary}

○ vlan-id — VLAN ID (1 to 4093)
○ primary — Bridge as primary or root bridge (primary bridge value is 24576)
○ secondary — Bridge as the secondary root bridge (secondary bridge value is 28672)
```

### Configure root bridge as primary

```
sonic(config)# spanning-tree vlan 1 root primary
```

### Verify root bridge information

```
sonic# show spanning-tree

Spanning tree enabled protocol rapid-pvst with force-version rstp
VLAN 1
Executing IEEE compatible Spanning Tree Protocol
Root ID Priority 24577, Address 90b1.1cf4.a523
Root Bridge hello time 2, max age 20, forward delay 15
Bridge ID Priority 24577, Address 90b1.1cf4.a523
We are the root of VLAN 1
Configured hello time 2, max age 20, forward delay 15
Interface Designated
Name PortID Prio Cost Sts Cost Bridge ID PortID
-----+
Eth1/5 128.276 128 500 FWD 0 24577 90b1.1cf4.a523 128.276
Eth1/6 128.280 128 500 LRN 0 24577 90b1.1cf4.a523 128.280
Interface
Name Role PortID Prio Cost Sts Cost Link-type Edge
-----+
Eth1/5 Desg 128.276 128 500 FWD 0 AUTO No
Eth1/6 Desg 128.280 128 500 LRN 0 AUTO No
```

## Global Rapid-PVST parameters

All nonroot bridges accept the timer values on the root bridge.

|                     |                                                                                                                                                                    |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Forward-time</b> | Amount of time that is required for an interface to transition from the Discarding state to the Learning state or from the Learning state to the Forwarding state. |
| <b>Hello-time</b>   | Time interval within which the bridge sends BPDUs.                                                                                                                 |
| <b>Max-age</b>      | Length of time the bridge maintains configuration information before it refreshes information by recomputing the Rapid-PVST topology.                              |

- Modify the forward-time in seconds (4 to 30, default 15).

```
sonic(config) # spanning-tree vlan vlan-range forward-time seconds
```

- Modify the hello-time in seconds (1 to 10, default 2). With large configurations involving more numbers of ports, Dell Technologies recommends increasing the hello-time.

```
sonic(config) # spanning-tree vlan vlan-range hello-time seconds
```

- Modify the max-age in seconds (6 to 40, default 20).

```
sonic(config) # spanning-tree vlan vlan-range max-age seconds
```

## View Rapid-PVST global parameters

```
sonic# show spanning-tree
Spanning-tree Mode: RPVST

VLAN 100 - RSTP instance 2

RSTP (IEEE 802.1w) Bridge Parameters:
Bridge Identifier Bridge MaxAge Hello FwdDly Hold
hex sec sec sec cnt
f064abaaaaaaaa11 20 2 15

RootBridge Identifier RootPath DesignatedBridge Root Max Hel Fwd
Identifier Cost Identifier Port Age lo Dly
hex hex
f0643c2c306fc204 0 20 2 15

RSTP (IEEE 802.1w) Port Parameters:
Port Num Prio Rity Path Cost P2P Edge BPDU Guard Role State Designa- Designated
Num rity Cost Mac Port Filter Type
PortChannel4 128 2000 Y N N LOOP DISABLED DISABLED 0 0000000000000000
...
...
```

## Rapid spanning-tree protocol

Rapid Spanning-Tree Protocol (RSTP) is similar to STP, but provides faster convergence and interoperability with devices configured with STP and MSTP. RSTP is disabled by default. All enabled interfaces in L2 mode automatically add to the RSTP topology.

Configuring RSTP is a two-step process:

1. Ensure that the interfaces are in L2 mode.
2. Globally enable RSTP.

## Multiple spanning tree protocol

This section provides general information about the multiple spanning tree protocol (MSTP) feature implementation and running MSTP over Layer 2 MLAG networks in Enterprise SONiC.

The more advanced implementation of STP - Rapid-PVST (RPVST) allows a spanning-tree instance for each VLAN. Running an RSTP instance for each VLAN may cost bandwidth and processing resources. MSTP mitigates this issue by configuring Multiple Spanning-Tree Instances (MSTIs) and mapping multiple VLANs to one spanning-tree instance in order to reduce the total number of STP instances.

## MSTP over MLAG

Enterprise SONiC supports MSTP on MLAG topologies. MSTP supports MLAG peers in both root bridge and non-root bridge roles in the network. MSTP on MLAG supports dual-homed deployments.

**i** **NOTE:** When you use MSTP over MLAG, configure both MLAG peers with the same MSTP configuration. For example, configure both MLAG peers with the same bridge priority value and on the same MST region. .

## Configure MSTP

Enable the spanning tree mode for the switch. To enable Multiple Spanning Tree (based on IEEE 802.1s), use the `mst` option.

```
sonic(config)# spanning-tree mode mst
```

**i** **NOTE:** You can enable only one of the spanning-tree modes at a time.

### Enter the MST configuration mode

This mode allows configuring the MST name, revision number, and the VLAN to instance mapping. All these parameters are used in determining the MST region to which the device belongs. The `no` form of the command removes the name, revision number, and the VLAN to instance mapping information and sets to the default values.

```
sonic(config)# spanning-tree mst configuration
sonic(config-mst) #
```

### Configure the region name for the MST

The default is the system MAC address.

```
sonic(config-mst) # name name-string
```

Enter a name for up to 32 characters. You can enter a null string by entering inverted commas.

### Configure the revision number for the MST

```
sonic(config-mst) # revision revision-number
```

*revision-number* - The range is from 0 to 65535. The default value is 0. By default, all VLANs are assigned to instance 0.

**i** **NOTE:** Changing the VLAN-to-MST-instance-mapping results in protocol reconvergence. Configure this mapping before enabling MSTP to avoid traffic disruption.

### Assign VLANs to an MST instance

```
sonic(config-mst) # instance id vlan id-or-range
```

**i** **NOTE:** The `no spanning-tree vlan` command has no impact on the MSTP configuration. If you run MSTP, Dell Technologies recommends that you do not use the `no spanning-tree vlan` command.

### Activate the configuration under the MST configuration mode

Use this command to activate the configuration under the MST configuration mode. Until this command is used, any modified configurations, such as, MSTP name, region, and instance to VLAN mapping does not take effect. This helps in ensuring all the devices in the network are configured with appropriate MSTP configurations before they take effect thus reducing the frequent reconvergences.

```
sonic(config-mst) # activate
```

### Abort the pending configurations that are not yet activated under the MST configuration mode

```
sonic(config-mst) # abort
```

The configurations that have already been activated are not affected by the abort operation.

### Configure the forward delay time

```
sonic(config)# spanning-tree mst forward-time seconds
```

Configure the forward delay time in seconds. The default value is 15. The range is from 4 to 30.

### Configure the hello interval for transmission of BPDUs

The default value is 2. The range is from 1 to 10.

```
sonic(config)# spanning-tree mst hello-time seconds
```

### Configure the maximum time to listen for the root bridge

The default value is 20 seconds. The range is from 6 to 40.

```
sonic(config)# spanning-tree mst max-age seconds
```

### Configure the maximum hops value for the MST

For MSTP, the maximum hops specifies the number of hops in a region before the BPDU is discarded and the information is aged out. The default value is 20. The range is from 1 to 40.

```
sonic(config)# spanning-tree mst max-hops value
```

### Configure the bridge priority in increments of 4096

The default value is 32768. The range is from 0 to 61440.

```
sonic(config)# spanning-tree mst mst-id-or-range priority priority-value
```

### Configure loop guard on all ports

By default, when spanning-tree stops receiving the BPDUs on a blocking port, it transitions to the forwarding state which can result in a loop in the network. The loop guard feature, when enabled, avoids this transition of nondesignated ports to forwarding state and instead moves the port to a "Loop inconsistent" state where the port continues to block the traffic to avoid the loop. By default, loop guard is disabled.

```
sonic(config)# spanning-tree loopguard default
```

### Configure the BPDU filter at the global level

BPDU filter is applied only if the port is in operational edge port mode. When global BPDU filter is enabled, around 10 BPDUs are sent at link up after which the port stops transmitting the BPDUs. If BPDUs are received, the port transitions from edge port to normal STP port and disables filtering for port to operate like a normal STP port. By default, BPDU filter is disabled.

```
sonic(config)# spanning-tree edge-port bpdufilter default
```

### Configure root guard

Root Guard provides a way to enforce the root bridge placement in the network and allows STP to interoperate with user network bridges while still maintaining the bridged network topology that the administrator requires. When BPDUs are received on a root guard-enabled port, the STP state is moved to "Root inconsistent" state to indicate this condition. Once the port stops receiving superior BPDUs, Root Guard automatically sets the port back to a FORWARDING state after the timeout expires.

You can configure the root guard globally on all switch ports or on a specified port interface. Specify a timeout value in seconds (5 to 600; default 30)

```
sonic(conf-if-Ethslot/port)# spanning-tree guard root {timeout seconds}
```

For example:

```
sonic(config)# spanning-tree guard root timeout 6
```

```
sonic(config-if-Ethernet0)# spanning-tree guard root
```

### Configure BPDU guard

The BPDU Guard feature disables the connected device's ability to initiate or participate in STP on edge ports. When STP BPDUs are received on the port where BPDU guard is enabled, the port shuts down. You can reenable the port administratively after ensuring the BPDUs have stopped coming on the port. You can configure the STP BPDU guard on an Ethernet or port-channel interface.

```
sonic(conf-if-Ethslot/port)# spanning-tree bpduguard [port-shutdown]
```

By default, BPDU guard only generates a syslog indicating the condition. To disable the port, use the `port-shutdown` option.

### Enable or disable sending or receiving of BPDUs on an interface

You can configure the STP BPDU filter on an Ethernet or port-channel interface.

```
sonic(conf-if-Ethslot/port) # spanning-tree bpdufilter {disable | enable}
```

**i** **NOTE:** The BPDU filter configuration on the interface takes precedence over the global BPDU filter configuration.

### Configure the port-level priority

To apply the port priority value on all MST instances on a port, configure the port priority without specifying an MST instance. Specify an MST ID to configure port priority only on one instance. The range is from 0 to 240. The default value is 128. You can configure the STP port priority on an Ethernet or port-channel interface.

```
sonic(conf-if-Ethslot/port) # spanning-tree [mst mst-id-or-range] port-priority priority-value
```

### Configure the port-level cost value

When configured without specifying the MST instance, cost value is applied to all the MST instances on the port. Specify an MST ID to configure cost only on one instance. The range is from 1 to 200000000. You can configure the STP cost on an Ethernet or port-channel interface.

```
sonic(conf-if-Ethslot/port) # spanning-tree [mst mst-id-or-range] cost cost-value
```

### Configure the link type

The default value is point-to-point. You can configure the STP link type on an Ethernet or port-channel interface.

```
sonic(conf-if-Ethslot/port) # spanning-tree link-type {point-to-point | shared}
```

### Configure the port type as edge

You can configure an Ethernet or port-channel interface as an edge port.

```
sonic(conf-if-Ethslot/port) # spanning-tree port type edge
```

## View MSTP information

```
sonic# show spanning-tree
#####
MST0 Vlans mapped: 1,3-99,101-199,201-4094
Bridge Address 8000.80a2.3526.0c5e
Root Address 8000.80a2.3526.0c5e
 Port Root Path cost 0
Regional Root Address 8000.80a2.3526.0c5e
 Internal cost 0 Rem hops 20
Operational Hello Time 2, Forward Delay 15, Max Age 20, Txholdcount 6
Configured Hello Time 2, Forward Delay 15, Max Age 20, Max Hops 20

Interface Role State Cost Prio.Nbr Type

Ethernet26 DESIGNATED FORWARDING 2000 128.26 P2P
Ethernet45 DESIGNATED FORWARDING 800 128.45 P2P
PortChannel120 DESIGNATED FORWARDING 2000 128.85 P2P

#####
MST1 Vlans mapped: 2,100,200
Bridge Address 8001.80a2.3526.0c5e
Root Address 8001.80a2.3526.0c5e
 Port Root Path cost 0 Rem hops 20

Interface Role State Cost Prio.Nbr Type

```

```

Ethernet45 DESIGNATED FORWARDING 800 128.45 P2P
PortChannel120 DESIGNATED FORWARDING 2000 128.85 P2P

```

```

sonic# show spanning-tree mst configuration
Name [Region1]
Revision 1 Instances configured 2
Instance Vlans mapped
Instance Vlans mapped

0 1,3-99,101-199,201-4094
1 2,100,200

```

```

sonic# show spanning-tree mst 1
MST1 Vlans mapped: 2,100,200
Bridge Address 8001.80a2.3526.0c5e
Root Address 8001.80a2.3526.0c5e
 Port Root Path cost 0 Rem hops 20
Interface Role State Cost Prio.Nbr Type
----- ----- ----- ----- ----- -----
Ethernet45 DESIGNATED FORWARDING 800 128.45 P2P
PortChannel120 DESIGNATED FORWARDING 2000 128.85 P2P
```

```

sonic# show spanning-tree mst 1 detail
MST1 Vlans mapped: 2,100,200
Bridge Address 8001.80a2.3526.0c5e
Root Address 8001.80a2.3526.0c5e
 Port Root Path cost 0 Rem hops 20
Ethernet45 is DESIGNATED FORWARDING
Port info port id 45 priority 128 cost 800
Designated root Address 8001.80a2.3526.0c5e cost 0
Designated bridge Address 8001.80a2.3526.0c5e port id 45
Timers: forward transitions 0
Bpdus sent 91, received 0

PortChannel120 is DESIGNATED FORWARDING
Port info port id 85 priority 128 cost 2000
Designated root Address 8001.80a2.3526.0c5e cost 0
Designated bridge Address 8001.80a2.3526.0c5e port id 85
Timers: forward transitions 0
Bpdus sent 93, received 0
```

```

sonic# show spanning-tree mst interface Ethernet45
Link Type: P2P Bpdu filter: False
Boundary: internal Bpdu guard: False

Instance Role State Cost Pri.Nbr Vlans mapped
----- ----- ----- ----- -----
0 DESIGNATED FORWARDING 800 128.45 1,3-99,101-199,201-4094
1 DESIGNATED FORWARDING 800 128.45 2,100,200
```

```

sonic# show spanning-tree bpdu-guard
PortNum Shutdown Port shut
 Configured due to BPDU guard

Ethernet2 Yes No
Port-Channel12 No NA
```

```

sonic# show spanning-tree inconsistentports
Loop guard default: Disabled
```

```

----- INST Inconsistency State
----- 0 Root Inconsistent
```

## Clear MSTP information

Clear spanning tree counters

```
clear spanning-tree counters interface
```

Clear the current protocol detected and restart the protocol migration process.

```
clear spanning-tree detected-protocol interface
```

### Debug spanning tree

Use the following debug commands for enabling additional logging:

- `debug spanning-tree`
- `debug spanning-tree mst {id | all}`
- `debug spanning-tree interface`
- `debug spanning-tree event`
- `debug spanning-tree verbose`
- `debug spanning-tree bpdu {tx|rx}`
- `debug spanning-tree reset` - Enable or disable all the debug flags.

## Port monitoring

Port monitoring copies (mirrors) specified traffic on ingress interfaces to a local or remote destination port for analysis.

### Types of port monitoring

- Local port monitoring — Port monitoring is performed on the same switch. The switch forwards a copy of incoming and outgoing traffic from one port to another local port for analysis. Local port monitoring is also known as *SPAN* (Switched Port Analyzer).
- Encapsulated remote port monitoring — Port monitoring is performed over a Layer 3 network. The traffic from the source port is encapsulated and forwarded to the destination port on another switch. Encapsulated remote port monitoring is also known as *ERSPAN* (Encapsulated Remote SPAN).

### Port monitoring configuration

To configure port monitoring:

- Use a SPAN mirror session to send traffic to a local port destination.
- Use an ERSPAN mirror session to send traffic to a remote device using a GRE tunnel.
- Use a flow-based monitoring policy to select traffic for port monitoring. Only matched traffic is mirrored to a local or remote monitoring device in a SPAN or ERSPAN session.

## Create a mirror session

Use a monitoring policy to send a copy of network packets selected on one switch port, multiple switch ports, an entire VLAN, or port channel to a local or remote destination port with an attached monitoring device.

### Port monitoring — Configuration notes

- Port monitoring supports up to four mirroring sessions in each direction — send (transmit) and receive. A monitoring policy does not affect the switching of network traffic on a source port, VLAN, or port channel. Each session supports a single source port and a single destination port.
  - A destination port can be shared across multiple mirror sessions in any direction. Destination-port only sessions can also be created and used in flow-based mirroring — see [Flow-based port monitoring](#). L2/L3 configuration on the destination port is not supported.
  - Multiple source ports in a single session are not supported. To mirror multiple source ports to the same destination port, configure multiple sessions, each with a different source port and the same destination port.
  - If remote port monitoring is configured in both directions — send (transmit) and receive — only four mirror sessions are supported.
  - If remote port monitoring is configured in one direction — send or receive — eight mirror sessions are supported.

- If the mirror ASIC resources that are shared between different mirror sessions are not available, a newly configured mirror session changes to an inactive state. As soon as the resources become available, the session is restored to active state.
- Port monitoring does not mirror CPU-generated frames to a local or remote destination port.
- Port monitoring does not mirror VXLAN-encapsulated frames on a tunnel-originating node to the CPU port.
- On S5232F-ON, S5248F-ON, and S5296F-ON switches, port monitoring does not mirror VXLAN-encapsulated frames on a tunnel-originating node to the CPU port.
- Up to 108 source ports are supported.
- One destination port is supported per mirror session. A packet received on a VLAN or port-channel member port, which is configured for monitoring, is mirrored to the destination port even if the packet is eventually trapped or discarded. Packets that are sent by the switch are mirrored when you configure Transmit (Tx) mirroring.
- The bandwidth limit for a mirror session is the maximum supported bandwidth on a destination port.

**(i) NOTE:** Mirroring does not guarantee that all traffic from the source ports is received on the destination port. If more data is sent to the destination than it can support, some data may be lost.

## Configure a mirror session

### 1. Configure a local (SPAN) or remote (ERSPAN) mirror session.

- To configure a SPAN mirror session, enter a mirror session name (up to 24 characters). In mirror-session mode, configure the local destination port interface, source interfaces if different from the interface on which the policy is applied, and the direction in which mirrored traffic is sent (tx) or received (rx).

```
sonic(config)# mirror-session session-name
sonic(conf-mirror-mirror1)# destination Eth slot/port[/breakout-port] [source Eth
slot/port[/breakout-port] | PortChannel number] {direction {rx | tx | both}}
```

For example:

```
sonic(config)# mirror-session mirror1
sonic(conf-mirror-mirror1)# destination Eth1/1 source Eth1/2 direction rx
```

- To configure a SPAN mirror session for the CPU port, enter a mirror session name. In mirror-session mode, specify the CPU local destination port, source interfaces if different from the interface on which the policy is applied, and the direction in which mirrored traffic is sent (tx) or received (rx).

```
sonic(config)# mirror-session session-name
sonic(conf-mirror-mirror1)# destination CPU [source Eth slot/port[/breakout-port] | PortChannel number] {direction {rx | tx | both}}
```

For example:

```
sonic(config)# mirror-session mirror1
sonic(conf-mirror-mirror1)# destination CPU source Eth1/2 direction rx
```

- To configure an ERSPAN mirror session, enter a mirror session name. In mirror-session mode, configure the remote destination, source interfaces, and the direction in which mirrored traffic is sent.

```
sonic(conf-mirror-mirror2)# destination erspan [dst-ip dst_ip] [src-ip src_ip]
[dscp ip_dscp] [gre ip_gre] [ttl ip_ttl] [queue queue_val] [source {Eth slot/port[/breakout-port] | PortChannel portchannel-number}] {direction {rx | tx | both}}
```

- *dst-ip dst\_ip* — Destination IP address in A.B.C.D format
- *src-ip src\_ip* — Mirrors packets from the specified source IP address in A.B.C.D format.
- *dscp ip\_dscp* — Mirrors packets with the specified DSCP value.
- *gre ip\_gre* — Mirrors packets using a GRE tunnel IP address.
- *ttl ip\_ttl* — Mirrors source packets with the specified Time-to-live value.
- *queue queue\_val* — Mirrors source packets with the specified traffic-class Type-of-Service (ToS) value.
- *source* — Source Ethernet port or port channel number
- *direction* — Mirroring direction: Receive, transmit, or both.

For example:

```
sonic(config)# mirror-session mirror2
sonic(conf-mirror-mirror2)# destination erspan dst-ip 10.1.1.1 src-ip 11.1.1.1 dscp
10 ttl 10 gre 0x88ee queue 10 source Eth1/2 direction rx
```

## View mirror session configuration

```
sonic# show mirror-session
ERSPAN Sessions

Name Status SRC-IP DST-IP GRE DSCP TTL Queue Policer SRC-Port Direction

Mirror2 active 11.1.1.1 10.1.1.1 0x88ee 10 10 10
 Eth1/2 rx

SPAN Sessions

Name Status DST-Port SRC-Port Direction

Mirror1 active Eth1/1 Eth1/2 rx
```

## Flow-based port monitoring

Use a monitoring policy to send a copy of network packets selected on one switch port, multiple switch ports, an entire VLAN, or port channel to a local or remote destination port with an attached monitoring device.

To use flow-based port monitoring:

1. Create a mirror session for flow-based port monitoring; for example, see **Example: Configure mirror session for flow-based monitoring** below.
2. Classify (select) traffic for port monitoring by using ACLs or the L2, L3, or L4 fields in packet headers.
3. In a policy map, configure the mirror session for the classified flow.
4. Apply the monitoring policy on ingress interfaces — globally all switch interfaces, a specified interface, a VLAN, or a port channel.

### Classify traffic using modular ACLs

To classify traffic using modular ACLs:

1. Create an L2, IPv4, or IPv6 ACL to identify a traffic flow.

```
sonic(config)# {mac | ip | ipv6} access-list name
```

2. Add permit and deny rules to the ACL for L2 MAC, IPv4, or IPv6 traffic — see [Configure ACLs](#); for example:

```
Create IP ACL
sonic(config)# ip access-list mirror_v4_acl
sonic(conf-ipv4-acl)# seq 1 permit ip any 89.0.0.0/24 remark RULE_1
sonic(conf-ipv4-acl)# seq 2 permit ip any 89.0.1.0/24 remark RULE_2
sonic(conf-ipv4-acl)# seq 3 permit ip any 89.0.2.0/24 remark RULE_3
sonic(conf-ipv4-acl)# seq 4 permit ip any 89.0.3.0/24 remark RULE_4
sonic(conf-ipv4-acl)# seq 5 permit ip any 89.0.4.0/24 remark RULE_5
```

3. Create a classifier (class map) of match-type acl.

```
sonic(config)# class-map name match-type acl
```

Add the L2, IPv4, or IPv6 ACL to the class map to select flow traffic. Each class map uses only one ACL: L2 MAC, IPv4, or IPv6; for example:

```
Create class map for mirroring IPv4 traffic
sonic(config)# class-map mirror_v4_class match-type acl
sonic(conf-class-map)# match access-group ip mirror_v4_acl
```

**(i) NOTE:** For a class map to be considered active, the ACL must be already configured. If the ACL is not configured, the classifier is incomplete and inactive. The class-map configuration is saved, and no error is displayed. When you configure the ACL, the classifier becomes active and applies any actions configured a policy.

### Classify traffic using L2-L4 header fields

For more fine-grained traffic classification in a flow, use match statements on L2, L3, and L4 header field values. You can combine match criteria for fields in different headers. For example, you can specify source MAC Address, VLAN, destination IP address, and TCP flags to identify a flow for forwarding actions. ACLs do not support this level of detailed packet classification.

A class map is considered invalid if you configure mutually exclusive header fields, such an IPv4 and an IPv6 address, as match criteria. If you enter no L2-L4 match statements in a class map, the classifier matches any traffic by default.

To classify traffic using L2-L4 header fields:

1. Create a classifier (class map) of match-type fields match-all.

```
sonic(config)# class-map name match-type fields match-all
```

2. Add match statements to select packets based on L2, L3, and L4 header values; for example:

```
sonic(config)# class-map mirror_classmap match-type fields match-all
sonic(config-class-map)# match vlan 1001
sonic(config-class-map)# match destination-address mac host 00:01:00:11:00:11
sonic(config-class-map)# match destination-address ip 1.1.1.0/24
sonic(config-class-map)# match ip protocol tcp
sonic(config-class-map)# match tcp-flags syn rst
```

## Configure and apply port monitoring policy

1. Create a monitoring policy map to configure the forwarding actions to take on classified traffic. The policy-map name must begin with an alphanumeric character; 63 characters maximum. It can contain alphanumeric, hyphen (-), and underscore (\_) characters.

```
sonic(config)# policy-map name type monitoring
```

2. In policy-map flow configuration mode, add a class map to the policy. Enter a priority number (0-4095) to specify the order in which a class map is applied in the policy map to match traffic in the flow. A higher priority class map is processed before a lower priority.

```
sonic(config-policy-map)# class class-map-name priority number
sonic(config-policy-map-flow) #
```

3. In policy-map flow configuration mode, enter the mirror session to use on classified traffic — see [Create a mirror session](#).

```
sonic(config-policy-map-flow)# set mirror-session session-name
```

4. Repeat Steps 4 and 5 to enter additional class maps and mirror sessions in the monitoring policy map.
5. Apply a port monitoring policy map globally on all switch interfaces, a specified interface, a VLAN, or a port channel. To remove a policy from an interface, enter the no version of the command. You can apply monitoring policies only on ingress interfaces.

- Globally on all switch interfaces:

```
sonic(config)# service-policy type monitoring in policy-map-name
```

- On an interface or subinterface:

```
sonic(config)# interface Eth slot/port[/breakout-port] [.subinterface]
sonic(config-if-Eth)# service-policy type monitoring in policy-map-name
sonic(config-subintf-Eth)# service-policy type monitoring in policy-map-name
```

- On VLAN interfaces:

```
sonic(config)# interface Vlan vlan-id
sonic(conf-if-Vlan)# service-policy type monitoring in policy-map-name
```

- On port-channel interfaces:

```
sonic(config)# interface PortChannel portchannel-number
sonic(conf-if-po)# service-policy type monitoring in policy-map-name
```

## View monitoring policy configuration

```
sonic# show policy-map {name | type monitoring}
```

```
sonic# show policy-map type monitoring
Policy mon_policy_0 Type monitoring
 Description:
 Flow fields class_0 at priority 999
 Description:
```

```

 mirror-session ERSPAN_DestIP_50.1.1.2
Flow fields_class_1 at priority 998
 Description:
 mirror-session ERSPAN_DestIP_60.1.1.2
Flow fields_class_2 at priority 997
 Description:
 mirror-session ERSPAN_DestIP_50.1.1.2
Flow fields_class_3 at priority 996
 Description:
 mirror-session ERSPAN_DestIP_60.1.1.2
Applied to:
 Eth1/1 at ingress

```

### **View monitoring policy binding**

```
sonic# show service-policy summary [interface {Eth slot/port[/breakout-port] [.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch} type monitoring
```

```

sonic# show service-policy summary
Eth1/1
 qos policy qos_policy0 at ingress
monitoring policy mon_policy_0 at ingress
PortChannel100
 qos policy policy0 at egress
Vlan100
 forwarding policy pbr0 at ingress
CPU
 acl-copp policy copp at ingress

```

### **View monitoring policy binding and counters**

```
sonic# show service-policy {interface {Eth slot/port[/breakout-port] [.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch} type forwarding | policy-map name [Eth slot/port[/breakout-port] [.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch]
```

```

sonic# show service-policy policy-map mon_policy_0
Eth1/1
 Policy mon_policy_0 Type monitoring at ingress
 Description:
 Flow fields_class_3 at priority 996 (Active)
 Description:
 mirror-session ERSPAN_DestIP_60.1.1.2
 Packet matches: 0 frames 0 bytes
 Flow fields_class_2 at priority 997 (Active)
 Description:
 mirror-session ERSPAN_DestIP_50.1.1.2
 Packet matches: 0 frames 0 bytes
 Flow fields_class_1 at priority 998 (Active)
 Description:
 mirror-session ERSPAN_DestIP_60.1.1.2
 Packet matches: 0 frames 0 bytes
 Flow fields_class_0 at priority 999 (Active)
 Description:
 mirror-session ERSPAN_DestIP_50.1.1.2
 Packet matches: 0 frames 0 bytes

```

To clear policy counters, enter the command:

```
sonic# clear counters service-policy {interface {Eth slot/port[/breakout-port] [.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch} type monitoring | policy-map name [Eth slot/port[/breakout-port] [.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch]
```

#### **Example: Configure mirror session with CPU destination port for flow-based monitoring**

```
! Create mirror session
sonic# mirror-session 2
sonic(conf-mirror-mirror2)# destination CPU direction both
```

#### **Example: Configure mirror session with ERSPAN for flow-based monitoring**

```
! Create mirror session
sonic# mirror-session 2
sonic(conf-mirror-mirror2)# destination erSPAN dst-ip 10.0.10.1 src-ip 10.10.20.1
```

## **Port Security**

Port security protects a port by limiting the number of MAC learns on a user-specified port.

The `port-security enable` command enables port security on a specific interface. When you enable the port security feature on a port, MAC learning mode is set on the port.

#### **MAC learning limit configuration**

Using the MAC learning feature, you can set the maximum limit on the number of MAC addresses that can be allowed on an interface. Limiting the MAC addresses provides security from MAC flooding. When the maximum allowed MAC threshold is exceeded, the system generates a warning syslog notification, and a port security violation occurs.

 **NOTE:** You can disable the MAC learning limit to restore the default number of allowed MAC addresses per port.

#### **Port security violation**

A port security violation occurs when more than the allowed number of MAC addresses are learned on the port. You can configure the action that should be taken during violation. The supported violated mode option is `protect`. The `protect` mode disables the MAC learning on the port when the MAC addresses on the interface reach the specified threshold. In disabled learning mode, all packets with unknown source addresses on the port are dropped. To undo the action, clear the MAC entries below the specified threshold.

#### **Default port security configuration**

- Port security on a port—Disabled
- Maximum number of MAC addresses per port—1
- Violation mode—Protect

## **Configure port security**

Use the following commands to configure port security:

1. Enter the configuration mode.

```
sonic# configure terminal
```

2. Enter the port number of the interface to be configured.

```
sonic(conf)# interface Ethernet 1
sonic(conf-if-Eth1/1) #
```

3. Enable port security on an interface.

```
sonic(conf-if-Eth1/1) # port-security enable
```

 **NOTE:** To disable the port security feature on an interface, use the `no port-security enable` command.

4. Configure the maximum number of secure MAC addresses allowed on an interface after you enable port security on an interface. The maximum range is from 1 to 4097; default is 1.

```
sonic(conf-if-Eth1/1) # port-security maximum maximum
```

5. Configure the action to take when there is a security violation. When a MAC learn limit violation is detected, this violation action protects the port by dropping all packets with unknown source MAC addresses.

```
sonic(conf-if-Eth1/1) # port-security violation protect
```

#### Port Security configuration example

```
sonic# configure terminal
sonic(conf) # interface Eth1/1
sonic(conf-if-Eth1/1) # port-security enable
sonic(conf-if-Eth1/1) # port-security maximum 11
sonic(conf-if-Eth1/1) # port-security violation protect
```

#### View port security configuration

View information for all MAC-security-enabled ports after enabling port security on the ports.

```
sonic# show port-security
Secure Port isEnabled MaxSecureAddr FdbCount ViolationCount
SecurityAction

-
Eth1/1 Y 11 11 360 PROTECT
```

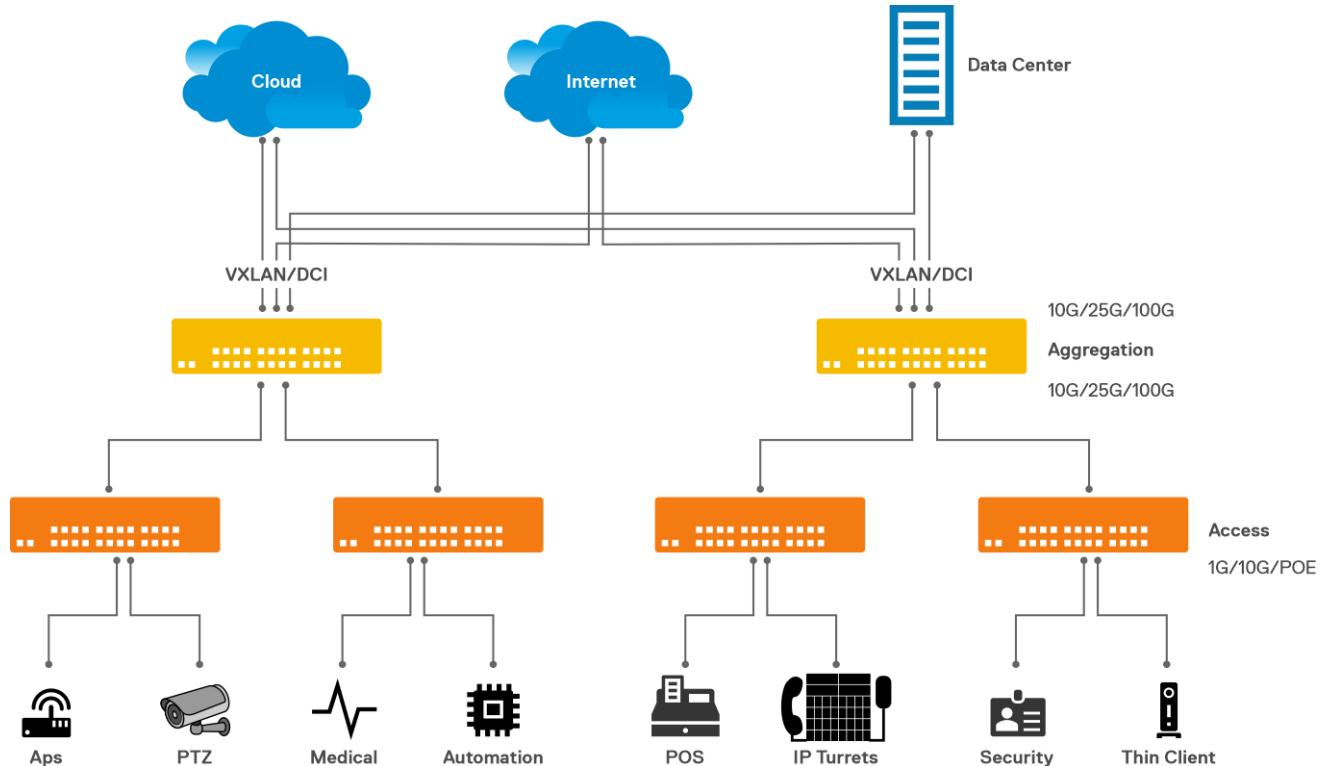
View interface-level port security information set on an interface.

```
sonic# show port-security interface Eth1/1
Interface : Eth1/1
Port MAC Security is Enabled : True
Maximum allowed Secure MAC : 11
Action taken on Violation : PROTECT
Total MAC address : 11
Security Violation Count : 360
```

## Layer 2 Edge features

Enterprise SONiC can serve as the unified network operating system (NOS) that you use to connect edge devices and a data center. The Edge bundle extends a data center fabric to remote locations using the same data center NOS.

- A two-layer fabric is implemented in which there is an aggregation and an access layer. Use VXLANs to stretch the fabric.
- The Clos network spine-and-leaf architecture allows for future scaling. Each leaf-layer access switch is connected to each spine-layer aggregation switch in a full-mesh topology.
- In the aggregation layer, VXLAN EVPN supports multi-tenancy and multi-site data center interconnection (DCI).
- Leverage the automation and management tools in the data center to configure and maintain edge switches.



**Figure 3. Layer 2 Edge networking**

In addition to many of the L2 features described in [Layer 2](#), Enterprise SONiC Edge bundle includes the following edge-specific features:

|                                               |                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Port access control (PAC)</b>              | Validates client and user credentials to prevent unauthorized access to a specific switch port (see <a href="#">Port access control</a> ), including: <ul style="list-style-type: none"> <li>• Dot1x authentication</li> <li>• MAC authentication bypass</li> <li>• Dynamic ACLs</li> <li>• RADIUS-provided VLANs</li> <li>• Dynamic Authorization Server</li> </ul> |
| <b>Power over Ethernet (PoE)</b>              | PoE provides power to connected devices — such as IP telephones, wireless LAN access points, and web cameras — over existing LAN cabling (see <a href="#">Power over Ethernet</a> ).                                                                                                                                                                                 |
| <b>Multiple Spanning Tree Protocol (MSTP)</b> | MSTP provides for efficient STP path redundancy when you use a large number of VLANs in a network. You can configure multiple VLANs in the same spanning-tree instance to reduce the number of instances needed to ensure only one active path between VLANs (see <a href="#">Multiple spanning tree protocol</a> ).                                                 |

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Reserved VLANs</b>        | The system reserves a certain number of VLANs for internal use. You can configure a different range of VLANs to be used as reserved VLANs (see <a href="#">Reserved VLANs</a> ).                                                                                                                                                                                                                                                                                        |
| <b>Voice VLANs</b>           | A dedicated VLAN for the user's voice data streams. A voice VLAN ensures the quality of voice traffic by improving the transmission priority when transmitted with other traffic (see <a href="#">LLDP-MED for voice VLANs</a> ).                                                                                                                                                                                                                                       |
| <b>Unauthenticated VLANs</b> | An unauthenticated VLAN allows access by both authorized and unauthorized devices or ports. Use an unauthenticated VLAN to authorize clients that time out or fail authentication due to invalid credentials. An unauthenticated VLAN must be a static VLAN, and cannot be the guest VLAN or the default VLAN. An unauthenticated VLAN applies only to 802.1x clients. Member ports must be manually configured as tagged members (see <a href="#">Special VLANs</a> ). |
| <b>Guest VLANs</b>           | A guest VLAN also authorizes clients that time out or fail authentication due to invalid credentials. A guest VLAN applies only to 802.1x unaware clients (see <a href="#">Special VLANs</a> ).                                                                                                                                                                                                                                                                         |

For information about how to configure and use VLANs, see [VLAN configuration](#).

### Topics:

- Port access control
- Power over Ethernet
- LLDP-MED for voice VLANs
- Reserved VLANs
- Multislot External Power Supply

## Port access control

The Port Access Control (PAC) feature provides validation of client and user credentials to prevent unauthorized access to a specific switch port.

Local Area Networks (LANs) are usually deployed in environments that permit unauthorized devices to be physically attached to the LAN infrastructure, or permit unauthorized users to attempt to access the LAN through equipment that are already attached. In such environments, restricting access to the services offered by the LAN to those users and devices (clients) that are permitted to use those services improves security. PAC provides a means for authenticating and authorizing devices that are attached to a LAN port that has point-to-point connection characteristics and of preventing access to that port in cases in which the authentication and authorization process fails.

PAC uses authentication methods, such as 802.1x and MAB for client authentication. These methods use RADIUS for verifying client credentials and receiving the authorization attributes, such as VLANs, ACLs, and so on, for the authenticated clients. The authentication session is initiated by the client.

### Dot1x

IEEE 802.1X-2004 is an IEEE Standard for Port Access Control (PAC) that provides an authentication mechanism to devices that want to attach to a LAN. The standard defines Extensible Authentication Protocol Over LAN (EAPOL). The 802.1X standard describes an architectural framework within which authentication and consequent actions take place. It also establishes the requirements for a protocol between the authenticator and the supplicant, as well as between the authenticator and the authentication server.

### MAC authentication bypass

An authenticator can use MAC Authentication Bypass (MAB) feature to authenticate simple devices like cameras or printers which do not support 802.1x. The MAB feature uses the device MAC address to authenticate the client.

### Scalability

- PAC supports 48 clients per port, with a maximum of 512 clients per switch.
- Supports 60 ACL rules per client/host.

## PAC target deployment

When a client authenticates itself initially on the network, the switch acts as the authenticator to the clients on the network and forwards the authentication requests to the RADIUS server in the network. If the authentication succeeds, the client is placed in authorized state and can forward or receive traffic through the port. RADIUS servers send a list of authorization attributes, such as VLAN and ACL configurations, to be applied to the client traffic. Port access control provides the flexibility to grant differential treatment to clients.

## Functional description

The Ports of an 802.1X Authenticator switch provide the means in which it can offer services to other systems reachable through the LAN. Port-based network access control allows the operation of the ports of the system to be controlled in order to ensure that access to its services is only permitted by systems that are authorized to do so.

PAC provides a means of preventing unauthorized access by Supplicants to the services offered by a system. Control over the access to a switch and the LAN to which it is connected can be desirable in order to restrict access to publicly accessible bridge ports or to restrict access to departmental LANs.

Access control is achieved by enforcing authentication of Supplicants that are attached to an Authenticator's controlled Ports. The result of the authentication process determines whether the Supplicant is authorized to access services on that controlled port.

A Port Access Entity (PAE) can act as an Authenticator, Supplicant, or Authentication server. Enterprise SONiC supports the Authenticator role only in which the PAE is responsible for communicating with the Supplicant. The Authenticator PAE is also responsible for submitting the information that is received from the Supplicant to the Authentication Server in order for the credentials to be checked, which determines the authorization state of the port. The Authenticator PAE controls the authorized or unauthorized state of the controlled port depending on the outcome of the authentication process.

## RADIUS authentication

The Authenticator acts as a passthrough for the EAP method. The Supplicant and the RADIUS server exchange EAP messages which are encapsulated in either EAPOL or RADIUS frames (depending on the direction of the frame) by the Authenticator switch. The Authenticator determines the authorization status of the port based on RADIUS Access-Accept or Access-Reject frames. The Authenticator switch also sends and processes all appropriate RADIUS attributes.

Enterprise SONiC works with Cisco ISE and FreeRADIUS for RADIUS authentication.

## Direction control for unauthenticated packets

The Authenticator controls the degree to which data traffic flow is controlled for an unauthenticated client. For example, an unauthorized controlled Port may exert control over communication in both directions (disabling both incoming and outgoing frames) or only in the incoming direction (disabling only the reception of incoming frames).

The control directions are of two types:

1. Both: Control is exerted over both incoming and outgoing frames.
2. In: Control is only exerted over incoming traffic.

Enterprise SONiC allows:

- "In" control: Only authorized clients are allowed to send traffic.
- "Out" control: Until the first client is authorized, all egress data traffic is blocked. After the first client is authorized, all egress traffic is allowed on the client VLAN on the port.

## Dynamic ACLs

Once a client on an access-controlled port is authenticated, dynamic ACLs (DACLs) allow an external RADIUS server to send ACL attributes based on the configured user profile on the RADIUS server. IPv6 and IPv4 ACLs are supported in a DACL. The ACL rules per client are sent in extended ACL syntax style. The switch applies the client-specific DACL during the authenticated session.

The switch does not display RADIUS specified DACL's in the running configuration. The ACL however shows up in the user interface show commands. The ACL configuration is only applied on the client-connected-port for the duration of the authenticated client session and is not persistent. The ACLs that are sent by RADIUS are in extended ACL syntax style and are validated as user-created ACLs. The Dynamic ACLs are managed by the PAC application of the operating system and you cannot delete them.

Generally, any static ACLs (created by the user) that are applied on the port are removed before applying the dynamic ACL on the port. Once the application-created dynamic ACL is removed or deleted, the static ACL is reapplied on the port. Essentially, static ACLs and dynamic ACLs are mutually exclusive. However if Open Authentication is configured on the port, the static ACLs and dynamic ACLs co-exist on the port. In such situations, the static ACLs have lower priority than the dynamic ACLs that are attached on the port. In situations where the client IP address changes, PAC learns about it through DHCP snooping binding tables and the application-created ACLs are automatically updated to accommodate the operational change such as a changed client IP address.

#### **Named ACLs**

The RADIUS server can also provide an attribute (filter name or filter ID) for PAC to apply a pre-configured ACL on the switch to the client. These preconfigured ACLs are named ACLs, and are created by the user on the switch. Once RADIUS indicates a named ACL is to be applied for a client, PAC replicates the ACL rules, modifies the rules to incorporate the client IP and then provides them as dynamic ACL rules.

## Downloadable ACLs

RADIUS server can also provide the ACL name - rules for which are defined on RADIUS server. This ACL name is provided by a Cisco AVP CiscoSecure-Defined-ACL. These ACLs are called as Downloadable ACLs. PAC downloads the ACL rules from the RADIUS server using the ACL name as username in the second Access-Request.

## RADIUS-supplied VLANs

PAC supports access control with the ability to control user profiles from a RADIUS server. Once a client is authenticated, the client authorization parameters from RADIUS can indicate VLAN association for client traffic. The VLAN that is associated with the client is a pre-created VLAN on the switch (static VLAN). If RADIUS does not supply a VLAN, the client is authorized on the configured Access VLAN of the port.

 **NOTE:** If the client sends traffic tagged with any other VLAN other than the one assigned to it, the packets are associated with the VLAN assigned by PAC.

## Special VLANs

The system keeps trying the next configured authentication method if the authentication method fails or times out. However if the last authentication fails, the system authorizes the client to special VLANs, namely, unauthenticated VLANs, or guest VLANs. If a client is authenticated in any of these VLANs and the VLAN is reconfigured or deleted from the system, all these clients are unauthorized.

#### **Unauthenticated VLAN**

An unauthenticated VLAN is used to authorize clients which fail authentication due to invalid credentials, and is used only for 802.1x-aware clients.

#### **Guest VLAN**

This is a special VLAN used to authorize 802.1X unaware clients.

## Host modes

#### **Single-host mode**

In this mode, only one data client is authenticated on a port and the client is granted access to the port. Access is allowed only for this client. Only when this client logs off, another client can get authenticated, authorized on the port, and granted port access.

#### **Multiple hosts mode**

In this mode, only one data client is authenticated on a port. However once authentication succeeds, access is granted to all clients connected to the port. Typical use case is a wireless access point which is connected to an access-controlled port of a NAS. Once the access point is authenticated by the NAS, the port is authorized for traffic from not just the access point but also from all the wireless clients connected to the access point. Once this client gets authenticated, the port is open for all clients connected to the port.

#### **Multiple-domain authentication mode**

In this mode, one data client and one voice client are authenticated on a port and these clients are then granted access. Typical use case is an IP phone connected to a NAS port and a laptop connected to the hub port of the IP phone. Both the devices need to be authenticated to access the network services behind the NAS. The voice and data domains are segregated. The RADIUS server attribute `Cisco-AVPair = "device-traffic-class=voice"` is used to identify a voice client.

#### **Multiple authentication mode**

In this mode, one voice client and multiple data clients are authenticated on a port and these clients are then granted access. Typical use case is a network of laptops and an IP phone connected to the NAS port through a hub.

## Redirect ACLs and URLs

A RADIUS server can provide `redirect-acl` and `redirect-url` attributes. Redirect ACLs are used to permit matching packets for HTTP redirection to the authenticating client and deny matching packets for forwarding. The ACL may be an existing named ACL or a downloadable ACL.

The redirect feature works in conjunction with redirect ACLs installed by PAC to redirect incoming client HTTP and HTTPS traffic to a new URL. The redirect ACLs have rules to trap matching packets. These packets are typically HTTP (TCP port 80) and HTTPS (TCP port 443) packets. Non matching packets follow their intended path. These rules are per client and client IP address is one of the matching criteria for the ACL.

The HTTPS connections require an SSL certificate which is a digital certificate that authenticates the server and enables an encrypted connection. SONiC supports configuration of X.509 certificate and associated keys through the "DEVICE\_METADATA|x509" CONFIG\_DB entry. .

## Configure port access control

Use the following commands to configure port access control.

- Enable authentication monitor mode on the switch. Use the monitor mode to help troubleshoot port-based authentication configuration issues without disrupting network access for hosts that are connected to the switch. In Monitor mode, a host is granted network access to an authentication-enforced port even if it fails the authentication process. The results of the process are logged for diagnostic purposes. By default, authentication monitor is disabled.

```
authentication monitor
```

- Enable dot1x authentication support on the switch. By default, dot1x authentication support is disabled.

```
dot1x system-auth-control
```

- Set the PAC role on the port.

```
dot1x pae {authenticator | none}
```

- authenticator - Sets the switch as the authenticator.
- none - Configures the switch to not process PAC (default).

- Configure a VLAN as a guest VLAN on an interface to authorize 802.1x-unaware clients which fail authentication or when their authentication times out.

```
authentication event no-response action authorize vlan vlan-id
```

- vlan `vlan-id` - The range is from 1 to the maximum that is supported on the platform. The default value is 0, that is, not configured.

- Configure the unauthenticated VLAN on an interface to authorize 802.1x-aware clients which fail authentication or when their authentication times out.

```
authentication event fail action authorize vlan vlan-id
```

- `vlan vlan-id` - The range is from 1 to the maximum that is supported on the platform. The default value is 0, which means the VLAN is not operational.
- Configure the number of retries by the client before a port moves to the authentication fail VLAN.

```
authentication event fail retry attempts
```

- `attempts` - The range is from 1 to 5. The default value is 3.

- Set the maximum number of clients that are supported on an interface when the multi-authentication host mode is enabled on the port.

```
authentication max-users count
```

- `count` - The range is from 1 to 48. The default value is 48.

- Enable periodic reauthentication of the supplicant for the specified interface. By default, this option is disabled.

```
authentication periodic
```

- Set the authentication mode to use on the specified interface. The default is `force-authorized`.

```
authentication port-control {auto | force-authorized | force-unauthorized}
```

- Configure the host mode of a port. The default is `multi-auth`.

```
authentication host-mode {multi-auth | multi-domain | multi-host | single-host}
```

- Configure the period of time after which the Authenticator attempts to reauthenticate a supplicant on the port. This command also provides an option to specify re-authentication time-out value from the RADIUS server. When the `server` option is selected, the server supplied session timeout and session termination-action are used by the Authenticator to reauthenticate a supplicant on the port. By default, `server` option is enabled. For reauthentication to happen after the configured or server provided timeout, enable the `authentication periodic` command. `server` is the default option.

```
authentication timer reauthenticate { seconds | server }
```

- `seconds` - The range is from 1 to 65535.

- Configure Open Authentication mode on the port. This option is disabled by default.

```
authentication open
```

- Set the order of authentication methods to be used on a port. If one method in the list is unsuccessful or timed out, the next method is attempted. Each method can only be entered once. The default order is `dot1x` and then `MAB`.

```
authentication order {dot1x [mab] | mab [dot1x]}
```

- Set the priority for the authentication methods to be used on a port. Authentication priority decides if the client that is already authenticated to re-authenticate with the higher-priority method when the same is received.

```
authentication priority {dot1x [mab] | mab [dot1x]}
```

- Enable MAC Authentication Bypass (MAB) on an interface. If enabled, EAP-MD5 is used by default.

```
mab [auth-type {pap | eap-md5} | chap]
```

- `pap` - Use PAP as the authentication method.
- `eap-md5` - Use EAP-MD5 as the authentication method.
- `chap` - Use CHAP as the authentication method.

- Set configuration parameters that are used to format attribute1 for MAB requests to the RADIUS server. RADIUS attribute 1 is the username, which is often the client MAC address. The default group size is 2. The default separator is `:`. The default case for MAC address is uppercase.

```
mab request format attribute 1 groupsize {1 | 2 | 4 | 12} separator {- | : | .}
[lowercase | uppercase]
```

- Configure `dot1x` timeout.

```
dot1x timeout {quiet-period quiet-period | server-timeout server-timeout}
```

- quiet-period *quiet-period* - Enter the value in seconds, which is the timer value that dot1x uses to define periods of time during which dot1x does not attempt to acquire a supplicant. This is the period for which the authenticator state machine stays in the HELD state. The default value is 2 seconds. The range is from 1 to 65535.
- server-timeout *server-timeout* - Enter the value in seconds, which is the timer value that dot1x uses to timeout the authentication server. The default value is 30 seconds. The range is from 10 to 65535.
- Configure MAB timeout.

```
mab timeout server-timeout server-timeout
```

- server-timeout *server-timeout* - Enter the value in seconds, which is the timer value that dot1x uses to timeout the authentication server. The default value is 30 seconds. The range is from 10 to 65535.

## Verify port access control

View the authentication manager information for the interface.

```
show authentication interface {all | { interface slot/port}}
```

```
sonic# show authentication interface Eth 1/1
```

```
Interface..... Eth1/1
Port Control Mode..... auto
Host Mode..... multi-auth
Open Authentication..... Disabled
Configured method order..... dot1x mab
Enabled method order..... dot1x mab
Configured method priority..... dot1x mab
Enabled method priority..... dot1x mab
Reauthentication Enabled..... TRUE
Reauthentication Session timeout from server .. FALSE
Reauthentication Period (secs)..... 15
Maximum Users..... 25
Guest VLAN ID..... 0
Unauthenticated VLAN ID..... 0
Authentication retry attempts..... 1
```

View the authentication manager global information and the number of authenticated clients.

```
show authentication
```

```
sonic# show authentication
```

```
Authentication Monitor Mode..... Disabled
Number of Authenticated clients..... 2
Number of clients in Monitor mode..... 0
```

View the details of the dot1x configuration.

```
show authentication clients {all | {interface slot/port}}
```

```
sonic# show authentication clients all
```

| Interface | MAC-Address       | Method | Host Mode  | Control Mode | VLAN   | Assigned Reason    |
|-----------|-------------------|--------|------------|--------------|--------|--------------------|
| Eth1/16   | 10:8D:B6:C6:00:00 | 802.1X | multi-host | auto         | RADIUS | Assigned VLAN (10) |

```
sonic# show authentication clients Eth 1/47
```

```
Mac Address 00:05:03:00:00:0A
User Name campus40
VLAN Assigned Reason Radius (105)
Host Mode multi-auth
```

```

Method 802.1x
Control Mode auto
Session time 95920
Session timeout (RADIUS) Not Applicable
Session timeout (Oper) 15
Time left for Session Termination Action Not Applicable
Session Termination Action Default
DACL ACSACL-IP-Dell_DACL1-63a2f506
Redirect ACL REDIRECTACL
Redirect URL
..... https://test.OS6:8443/portal/gateway?
sessionId=64680714Kj2IILS2eS5hTyK8ThFPiRY9ci03Z0WUFosxGQEgaHg&portal=71984f36-f55e-4439-
ba6e-903d9f77c216&action=cwa&token=ab9bcc0f6c417c42e00e56fd12148a0b

```

View the authentication manager authentication history log.

```
show authentication authentication-history {all | {interface slot/port}}
```

```
sonic# show authentication authentication-history Eth 1/2
```

| Timestamp            | Interface | MAC-Address       | Auth Status  | Method |
|----------------------|-----------|-------------------|--------------|--------|
| May 07 2020 13:02:41 | Eth1/2    | 58:05:94:1C:00:00 | Unauthorized | 802.1X |
| May 07 2020 13:01:33 | Eth1/2    | 58:05:94:1C:00:00 | Unauthorized | 802.1X |

View a summary of the global MAB configuration and summary information of the MAB configuration for all ports.

```
show mab [interface slot/port]
```

```
sonic# show mab
```

```

MAB Request Fmt Attr1 Groupsize 2
MAB Request Fmt Attr1 Separator legacy(:)
MAB Request Fmt Attr1 Case uppercase

```

```

Interface Eth1/10
Admin mode Disabled
mab_auth_type EAP_MD5
Server Timeout(secs) 30

```

```

Interface Eth1/11
Admin mode Disabled
mab_auth_type EAP_MD5
Server Timeout(secs) 30

```

```

sonic #show mab interface Eth 1/10
Interface Eth1/10
Admin mode Enabled
mab_auth_type EAP_MD5
Server Timeout(secs) 60

```

View a summary of the global dot1x configuration.

```
show dot1x
```

```
sonic# show dot1x
```

```
Administrative Mode Enabled
```

### **Clear commands**

Clear information for all Auth Manager sessions. All the authenticated clients are reinitialized and forced to authenticate again.

```
clear authentication sessions interface {all | Eth slot/port}
```

Clear authentication sessions of a MAC address.

```
clear authentication sessions mac mac-address
```

Clear authentication history log.

```
clear authentication history interface {all | Eth slot/port}
```

## Configure URL redirection

To configure URL redirection, follow this procedure:

- Configure a new security profile for the redirect feature.

```
redirect security-profile security-profile-name
```

The security profile points to the CA certificate file for redirecting HTTPS connections.

## Power over Ethernet

Power over Ethernet (PoE) technology allows IP telephones, wireless LAN access points, web cameras, and many other appliances to receive power and data over the existing LAN cabling, without needing to modify the existing Ethernet infrastructure.

## Target use cases for PoE

With PoE, you can perform following actions:

- Provide power to requesting devices attached directly to the switch.
- Prevent some or all PoE ports from delivering power.
- Manage the amount of power that can be delivered on a PoE port.
- View the electrical measurements and power delivery status of the PoE ports.
- Restore PoE port to normal state when it is in a fault state.

**(i) NOTE:** PoE features are supported only on N3248PXE-ON, E3248PXE-ON, and E3248P-ON switches.

You can enable PoE only on copper ports. However, depending on the platform, all or some of the available copper ports can be eligible for PoE functionality.

## Supported PoE specifications

### 802.3af and legacy support

The PoE enabled network switches intending to supply power (PSE) provide the PSE functionality as specified in IEEE 802.3af specification. The devices drawing power (PD) implement the PD functionality of IEEE 802.3af specification. The Enterprise SONiC PoE implementation conforms to the IEEE 802.3af PoE specification. In addition, legacy devices (non-IEEE 802.3af compliant) can also be powered up using the legacy detection feature.

### 802.3at - High power applications

Enterprise SONiC PoE uses the PoE+ specification (IEEE 802.3AT), which allows power to be supplied to Class 4 PD devices that require power greater than 15.4 Watts and up to 30 Watts. PoE-enabled network switches and routers can be deployed with devices that require more power than the dot3af specification.

### 802.3bt and pre-802.3bt support

Pre-802.3bt enables 60 Watts of power-on devices which support this feature.

The 802.3bt specification introduces Type 3 and Type 4 devices allowing power levels up to 51W for Type 3 PDs (60 W PSE) and up to 71.3W (90W PSE) for Type 4 PDs. Current is provided through all four twisted pairs in the network cable. The 802.3bt specification also introduces power classes 5-8.

### Flexible power management

The SONiC PoE solution provides power management which supports power reservation, power prioritization, and power limiting. Administrators can assign a priority to each PoE port. When the PoE switch has less power available and more ports are required to supply power, higher priority ports are receive power in preference to lower priority ports.

Lower priority ports are forcibly stopped to supply power in order to provide power to higher priority ports. In the Dynamic Power Management feature, power is not reserved for a given port at any point of time.

Class-based power mangement reserves a class-based amount of power for a PoE port.

The power that is available with the PoE switch is calculated by subtracting the instantaneous power drawn by all the ports from the maximum available power. Thus more ports can deliver power at the same time. This feature is useful to efficiently power up more devices when the available power with the PoE switch is limited.

## Configure PoE

### Enable PoE

By default, PoE is enabled on all ports. When enabled, PoE delivers power to an attached device. To disable PoE on a port or port range, enter the `poe disable` command. To re-enable PoE, enter the `no poe disable` command.

```
sonic(config)# interface Ethslot/port[/subport]
sonic(config-if-Eth)# no poe disable
```

### Set PoE detection mode

Use detection mode to set the type of devices that PoE can detect and power up. By default, PoE powers up both IEEE standard devices and pre-IEEE legacy devices which were pre-standard. If you restrict the PoE controller to detect only IEEE standard devices (`poe detection dot3bt`), you can return to the default detection setting by entering `no poe detection`.

```
sonic(config-if-Eth)# poe detection {dot3bt | dot3bt+legacy}
```

- `dot3bt` - PoE detects only IEEE standard devices.
- `dot3bt+legacy` - PoE detects IEEE standard devices and pre-IEEE legacy devices (default).

### Configure PoE port priority

Sometimes the switch may not be able to supply power to all connected devices. The port priority is used to determine which ports supply power if adequate power capacity is not available for all PoE-enabled ports. If ports are configured with the same priority level, a lower-numbered port has higher priority. By default, a PoE-enabled port has low priority.

When the switch is delivering peak power to a certain number of devices and you attach a new device to a high-priority port, power to a low-priority port is shut down and the new device is powered up.

```
sonic(config-if-Eth)# poe priority {critical | high | low}
```

### Set power management

Use the `poe power management` command to set the algorithm used by the PoE port to deliver power to requesting powered devices (PDs).

```
sonic(config)# poe power management {class | dynamic}
```

- `class` - Class-based power management.
- `dynamic` - Power management is performed by the POE controller. The maximum power provided by a port is not reserved for each port.

### Reset PoE port

Use the `poe reset` command to reset power-supply (PSE) operation on all PoE ports or on a specified PoE port. The port stops delivering power and performs the PoE detection and power delivery cycle again.

```
sonic# poe reset [Ethslot/port[/subport]]
```

### Clear PoE counters

Use the `clear poe counters` command to clear PoE error counters on a specified port or on all ports.

```
sonic# clear poe counters [Ethslot/port[/subport]]
```

## View PoE information

To display the current PoE configuration and the system-wide status information:

```
sonic# show poe

Firmware Version : 3.55
Total Power Available : 1056 Watts
Threshold Power : 950.0 Watts
Total Power Consumed : 97.0 Watts
Usage Threshold : 90 %
Power Management Mode : Dynamic
```

### View the PoE port configuration

```
sonic# show poe port configuration {all | interface slot/port}
```

```
sonic# show poe port configuration all
```

| Port                           | Admin Mode | Priority | Power Limit (mW) | Power Limit Type | High Power | Power-Up Mode | Detection Type |
|--------------------------------|------------|----------|------------------|------------------|------------|---------------|----------------|
| Eth1/1                         | Enabled    | Low      | 99900            | Class Based      |            | dot3bt        | dot3bt+legacy  |
| Eth1/2                         | Enabled    | Low      | 99900            | Class Based      |            | dot3bt        | Dot3bt         |
| Eth1/3                         | Enabled    | Low      | 99900            | Class Based      |            | dot3bt        | dot3bt+legacy  |
| <output truncated for brevity> |            |          |                  |                  |            |               |                |
| Eth1/46                        | Enabled    | Low      | 99900            | Class Based      |            | dot3bt        | dot3bt+legacy  |
| Eth1/47                        | Enabled    | Low      | 99900            | Class Based      |            | dot3bt        | dot3bt+legacy  |
| Eth1/48                        | Enabled    | Low      | 99900            | Class Based      |            | dot3bt        | dot3bt+legacy  |

### View PoE port information

```
sonic# show poe port info {all | interface slot/port}
```

```
sonic# show poe port info Eth 1/48
```

| Port                      | Class Requested | Class Assigned | Output Power (mW) | Output Current (mA) | Output Voltage (V) | Temp (C) | Status     | Fault Status |
|---------------------------|-----------------|----------------|-------------------|---------------------|--------------------|----------|------------|--------------|
| Eth1/48                   | 2               | 2              | 2700              | 49                  | 56.1               | N/A      | Delivering | No Error     |
| Overload Counter : 0      |                 |                |                   |                     |                    |          |            |              |
| Short Counter             | :               | 0              |                   |                     |                    |          |            |              |
| Power Denied Counter      | :               | 0              |                   |                     |                    |          |            |              |
| Absent Counter            | :               | 0              |                   |                     |                    |          |            |              |
| Invalid Signature Counter | :               | 0              |                   |                     |                    |          |            |              |

```
sonic# show poe port info all
```

| Port                           | Class Requested | Class Assigned | Output Power (mW) | Output Current (mA) | Output Voltage (V) | Temp (C) | Status    | Fault Status        |
|--------------------------------|-----------------|----------------|-------------------|---------------------|--------------------|----------|-----------|---------------------|
| Eth1/1                         | Unknown         | Unknown        | 0                 | 0                   | 0                  | N/A      | Fault     | OOR Capacitor Value |
| Eth1/2                         | Unknown         | Unknown        | 0                 | 0                   | 0                  | N/A      | Fault     | OOR Capacitor Value |
| Eth1/3                         | Unknown         | Unknown        | 0                 | 0                   | 0                  | N/A      | Fault     | OOR Capacitor Value |
| Eth1/4                         | Unknown         | Unknown        | 0                 | 0                   | 0                  | N/A      | Searching | No Error            |
| <output truncated for brevity> |                 |                |                   |                     |                    |          |           |                     |

|         |         |         |       |      |      |     |            |    |       |
|---------|---------|---------|-------|------|------|-----|------------|----|-------|
| Eth1/42 | 8       | 8       | 94600 | 1696 | 55.8 | N/A | Delivering | No | Error |
| Eth1/43 | Unknown | Unknown | 0     | 0    | 0    | N/A | Searching  | No | Error |
| Eth1/44 | Unknown | Unknown | 0     | 0    | 0    | N/A | Searching  | No | Error |
| Eth1/45 | Unknown | Unknown | 0     | 0    | 0    | N/A | Searching  | No | Error |
| Eth1/46 | Unknown | Unknown | 0     | 0    | 0    | N/A | Searching  | No | Error |
| Eth1/47 | Unknown | Unknown | 0     | 0    | 0    | N/A | Searching  | No | Error |
| Eth1/48 | 2       | 2       | 2600  | 47   | 56   | N/A | Delivering | No | Error |

## LLDP-MED for voice VLANs

Link Layer Discovery Protocol-Media Endpoint Discovery (LLDP-MED) is an extension to LLDP to provide interoperability between endpoint devices such as VoIP, and other networking end-devices.

LLDP-MED supports the following TLVs:

- Network policy TLV
- Power management TLV

### Network policy TLV

The Network policy TLV allows the device to advertise the voice VLAN information to endpoint devices like VoIP phones. Along with voice VLAN, tagging mode, Dot1p CoS, and DSCP values can be sent to the endpoint device. The Voice VLAN feature enables switch ports to carry voice traffic with a defined priority to enable separation of voice and data traffic coming onto the port. The separation of voice traffic ensures that the sound quality of an IP phone is safeguarded from deteriorating when the data traffic on the port is high.

**(i)** **NOTE:** Enable voice VLAN only on compatible devices.

The following combinations of voice VLAN traffic are supported:

- Assign Voice VLAN to an IP phone — the most common deployment. The phone sends voice packets that are tagged with the voice VLAN; data traffic is sent untagged.
- Assign a Dot1p priority to an IP phone. The phone sends voice packets with the 802.1p tag and data traffic is untagged.
- Allow the IP phone to send untagged voice traffic. Voice traffic cannot be differentiated from data traffic and no QoS can be provided.
- LLDP does not automatically apply QoS policies or the VLAN configuration. User must configure appropriate QoS policies to prioritize traffic based on CoS and DSCP values, and also the VLAN membership on the interfaces for data and voice traffic.

### Power management TLV

Allows the network device and endpoint device to exchange the power information, such as how the device is powered on, power priority, power required by the device, and so on.

### LLDP 802.3 Power using MDI TLV

Supports power negotiation between network device and endpoint device using LLDP 802.3 power using the MDI TLV.

### LLDP Management TLV IP address

By default, LLDP advertises the management IP address that is configured in the system over the management interface. You can override this setting and advertise specific IPv4 and IPv6 address by configuring the IP addresses to be advertised by LLDP on an interface.

### Configure LLDP MED

To configure LLDP-MED, follow this procedure:

1. Create a network policy profile.

```
sonic(config) # network-policy profile profile-number
```

2. Configure the network policy profile parameters

```
sonic(conf-network-policy) # {voice | voice-signaling} vlan [vlan-id {[cos
cos-value | dscp dscp-value] | untagged }| **[dot1p {** cos cos-value
| dscp value}]]
```

- voice - Selects the application.
- voice-signaling - Selects the application type.
- vlan vlan-id - Specifies the voice VLAN.

- `cos cos-value` - Specifies the L2 priority class of service value for the VLAN.
- `dscp value` - Specifies the differentiated services code point (DSCP) value.
- `dot1p` - Allows the use of 802.1 priority tagging with VLAN 0.
- `untagged` - Specifies voice traffic to be untagged.

3. Configure an IPv4 or IPv6 management address that is used to advertise by LLDP on an interface.

```
sonic(conf-if)# lldp tlv-set { management-address {ipv4|ipv6} ip-address}
```

4. Configure in the interface whether to advertise the LLDP-MED TLVs or not. By default the LLDP-MED TLVs are advertised.

```
sonic(conf-if)#lldp med-tlv-select [network-policy | power-management]
```

- `network-policy` - Select to advertise network policy TLVs.
- `power-management` - Select to advertise power management TLVs

5. Apply the LLDP-MED network policy to an interface.

```
sonic(conf-if)# network-policy profile-number
```

### Example configuring LLDP-MED

```
sonic(config)# network-policy profile 1
sonic(config-network-policy)# voice vlan 100 cos 4 dscp 20
sonic(config)# interface Eth 1/1
sonic(config-if-Eth1/1)# network-policy 1
```

### View LLDP-MED information

The following example displays the LLDP-MED power management information of the endpoint:

```
sonic# show lldp neighbor Eth 1/48

LLDP Neighbors

Interface: Eth1/48, via: LLDP
Chassis:
 ChassisID: 30.0.0.3
 SysName: AVXECFD5B
 SysDescr:
 TTL: 120
 MgmtIP: 30.0.0.3
 Capability: MAC_BRIDGE, ON
Port
 PortID: 2c:f4:c5:ec:fd:5b
 PortDescr:
LLDP-MED
 Device Type: Communication Device Endpoint (Class III)
 Capability: Capabilities, yes
 Capability: Ext_mdi_power_pd, yes
 Capability: Inventory, yes
 Capability: Network_policy, yes
 LLDP-MED Network Policy for: Voice
 VLAN: 30
 Priority: 6
 DSCP: 42
 Extended Power-over-Ethernet
 Power Type: PD
 Source: PSE
 Priority: High
 Value: 0
```

The following example displays the LLDP 802.3 Power via MDI TLV information of the endpoint:

```
sonic# show lldp neighbor Ethernet 2

LLDP Neighbors

Interface: Ethernet2, via: LLDP
Chassis:
 ChassisID: 63.63.63.3
```

```

SysName: SEP00562BB45460
SysDescr: Cisco IP Phone 8851, v1, sip88xx.11-0-1-11.loads
TTL: 180
MgmtIP: 63.63.63.3
Capability: MAC_BRIDGE, ON
Port
PortID: 00562BB45460:P1
PortDescr: SW PORT
MDI Power: supported: no, enabled: no, pair control: no
Device type: PD
Power pairs: signal
Class: class4
Power type: Type2
Power Source: PSE
Priority: Unknown
Requested: 88
Allocated: 88

```

To display brief neighbor information:

```

sonic# show lldp table

----- LocalPort RemoteDevice RemotePortID Capability
----- RemotePortDescr
----- Eth1/48 AVXECFD5B 2c:f4:c5:ec:fd:5b B

```

The following example displays the LLDP statistics:

```

sonic# show lldp statistics Ethernet4
LLDP Statistics

Interface: Ethernet4
 Transmitted : 400
 Received : 394
 Discarded : 0
 Unrecognized TLV : 0
 Ageout : 0

```

## Reserved VLANs

Enterprise SONiC implements a range of reserved VLANs for use by various protocols.

When a feature requires one or more reserved VLANs, the feature uses a VLAN that is not already in use from the default range. The default reserved VLAN range is from 3967 to 4094.

If there are no free VLANs in the default VLAN range, the feature that requires reserved a VLAN may not function. To avoid this issue, perform one of the following actions:

- Free up a few VLANs from the default range.
- Change the reserved VLAN range.

When you migrate from an older release to Enterprise SONiC 4.0, any VLAN that is already present in the default reserved VLAN range continues to work and the software allows additional configuration on that VLAN. However, the software does not permit creating a new VLAN within the reserved VLAN range.

You can configure the reserved VLANs to any 128 continuous VLAN range.

### Change the default range of reserved VLANs

To change the default range of reserved VLANs, use the following procedure:

- Use the following command:

```
sonic(config)# system vlan vlan-range reserve
```

 **NOTE:** Use the no form of this command to revert to the default range.

#### Example: Configure reserved VLAN range

```
sonic(config)# system vlan 400 reserve
```

#### View reserved VLAN range

The following command displays the default or configured reserved VLAN range:

```
sonic(config)# show system vlan reserved
system vlan reservation: 400-527
```

## Multislot External Power Supply

The N3248PXE and E3248PXE platforms require more power than what the internal power supplies provide to support PoE on all ports. The Multislot External Power Supply (MEPS) functionality enables the N3248PXE and E3248PXE platforms to use two external power supplies.

The following list explains the MEPS feature:

- You can use a Single slot External PSU (SEPS) or a Multislot External PSU (MEPS).
- You cannot use both SEPS and MEPS simultaneously.
- When the system boots or reboots, ensure that internal PSUs are inserted.
- MEPS consists of six ports—1 A, 1 B, 2 A, 2 B, 3 A, and 3 B. The A ports include power path, a signal pin, and a stand-alone PMBus pin. The B ports only have the present pin and a 54V power path.
- To receive power from MEPS, connect cables on both A and B ports.
- Each port supplies 45A of current enabling a maximum of 90A of current.
- When both Port A and Port B are connected to the switch system, the maximum power that is provided for one output slot is 4800 W.
- If an EPS is connected to the switch and not powered on, the `show platform psusummary` and `show platform psustatus` commands display the PSU status as NOT PRESENT.

#### Monitor EPS system health

By default, the system ignores the EPS (PSU 3) from the system health monitor list. To monitor the EPS, modify the `system_health_monitoring_config.json` file.

On the N3248PXE-ON platform, modify the `/usr/share/sonic/device/x86_64-dell EMC_n3248pxe_c3338-r0/system_health_monitoring_config.json` file.

On the E3248PXE-ON platform, modify the `/usr/share/sonic/device/x86_64-dell_e3248pxe-r0/system_health_monitoring_config.json` file.

The following example shows modifying the `system_health_monitoring_config.json` file:

```
root@N3248PXE:~# cat /usr/share/sonic/device/x86_64-dell EMC_n3248pxe_c3338-r0/system_health_monitoring_config.json
{
 "services_to_ignore": [],
 "devices_to_ignore": ["fan.speed", "psu.temperature", "psu.voltage", "asic", "PSU 3", "PSU 3 FAN 1"],
 "user_defined_checkers": [],
 "polling_interval": 60,
 "led_color": {
 "fault": "blink_yellow",
 "normal": "green",
 "booting": "blink_green"
 }
}
```

#### View MEPS information

The show platform psusummary command displays N/A for Input Current, Input Power, and Input Voltage on the N3248PXE-ON and E3248PXE-ON platforms for external PSUs.

```
sonic# show platform psusummary
PSU 1:
Description :DPS-1600AB-34 C
Fans :1
Mfg Name :DELTA
Name :PSU 1
Oper Status :OK
Serial Number :xxxxxxxxxxxx
Status LED :None
Type (AC/DC) :AC
Input Current (A) :3.00
Input Power (W) :669.00
Input Voltage (V) :225.80
Output Current (A) :11.30
Output Power (W) :629.00
Output Voltage (V) :55.60
Fan Speed (RPM) :8352
Fan Direction :exhaust
Temperature :35.40
PSU 2:
Description :DPS-1600AB-34 C
Fans :1
Mfg Name :DELTA
Name :PSU 2
Oper Status :OK
Serial Number :xxxxxxxxxxxx
Status LED :None
Type (AC/DC) :AC
Input Current (A) :3.00
Input Power (W) :681.00
Input Voltage (V) :225.20
Output Current (A) :11.70
Output Power (W) :646.00
Output Voltage (V) :55.50
Fan Speed (RPM) :7920
Fan Direction :exhaust
Temperature :33.20
PSU 3:
Description :04JR64
Fans :0
Mfg Name :DELTA
Name :PSU 3
Oper Status :OK
Serial Number :xxxxxxxxxxxxxx
Status LED :None
Type (AC/DC) :Unknown
Input Current (A) :N/A
Input Power (W) :N/A
Input Voltage (V) :N/A
Output Current (A) :22.90
Output Power (W) :1276.00
Output Voltage (V) :55.70
Fan Speed (RPM) :0
Fan Direction :none
Temperature :N/A
```

## Layer 3

|                                                  |                                                                                                                                                                                                                                                                                                |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Virtual routing and forwarding (VRF)</b>      | Provides a mechanism to partition a physical router into multiple virtual routers (see <a href="#">Virtual routing and forwarding</a> ).                                                                                                                                                       |
| <b>Bidirectional forwarding detection (BFD)</b>  | Provides rapid failure detection in links with adjacent routers (see <a href="#">Virtual routing and forwarding</a> ).                                                                                                                                                                         |
| <b>Border gateway protocol (BGP)</b>             | Provides an external gateway protocol that transmits inter-domain routing information within and between autonomous systems (see <a href="#">Border gateway protocol</a> ).                                                                                                                    |
| <b>Address resolution protocol (ARP)</b>         | Enables IPv4 packets to be sent across networks by translating IP network addresses to MAC hardware addresses, and MAC addresses to IP addresses. View the MAC address and corresponding IP address of destination devices in the ARP table; see <a href="#">Address resolution protocol</a> . |
| <b>Open shortest path first (OSPF)</b>           | Provides a link-state routing protocol that communicates with all other devices in the same autonomous system using link-state advertisements (see <a href="#">Open shortest path first</a> ).                                                                                                 |
| <b>Route-maps</b>                                | Defines which route are allowed to be redistributed into the target routing process (see <a href="#">Route-maps</a> ).                                                                                                                                                                         |
| <b>IPv4 and IPv6 static routes</b>               | Configures fixed, static routes to ensure that routed traffic can be exchanged with a specified destination device (see <a href="#">Static routes</a> ).                                                                                                                                       |
| <b>Virtual router redundancy protocol (VRRP)</b> | Provides a mechanism to eliminate a single point of failure in a statically routed network (see <a href="#">Virtual router redundancy protocol</a> ).                                                                                                                                          |
| <b>Network Address Translation (NAT)</b>         | Assigns a public IP address to switches that access resources outside the network (see <a href="#">Network Address Translation</a> ).                                                                                                                                                          |

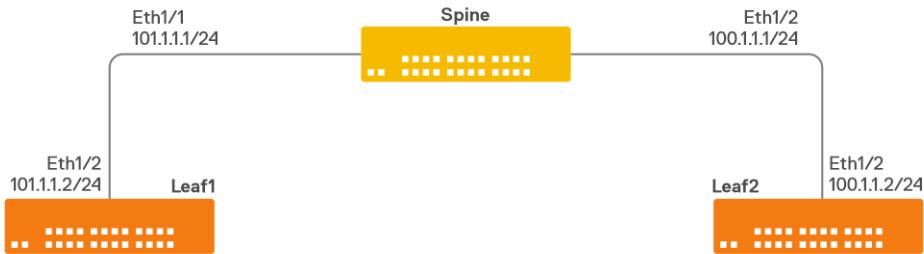
### Topics:

- Configure IPv4 and IPv6 address
- Virtual routing and forwarding
- Border Gateway Protocol
- IPv4 Address Resolution Protocol
- Open Shortest Path First
- Route-maps
- Static routes
- View IP routes
- Policy-based routing
- Virtual Router Redundancy Protocol
- Network Address Translation
- ECMP
- IP helper

## Configure IPv4 and IPv6 address

### IPv4 address configuration

An IPv4 address configuration on spine and leaf switch interfaces with no additional routing protocols is shown here:



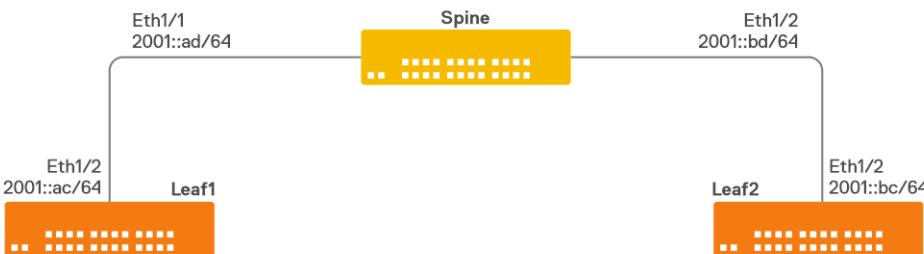
**Figure 4. IPv4 address configuration**

**Table 32. IPv4 address configuration**

| Spine configuration                                                                                                                                                            | Leaf1 configuration                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>sonic# configure sonic(config)# interface Eth1/1 sonic(config-if-Eth1/1)# no shutdown sonic(config-if-Eth1/1)# ip address 101.1.1.1/24 sonic(config-if-Eth1/1)# end</pre> | <pre>sonic# configure sonic(config)# interface Eth1/2 sonic(config-if-Eth1/1)# no shutdown sonic(config-if-Eth1/1)# ip address 101.1.1.2/24 sonic(config-if-Eth1/1)# end</pre> |

#### IPv6 address configuration

An IPv6 address configuration on spine and leaf switch interfaces with no additional routing protocols is shown here:



**Figure 5. IPv6 address configuration**

**Table 33. IPv6 address configuration**

| Spine configuration                                                                                                                                                             | Leaf1 configuration                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>sonic# configure sonic(config)# interface Eth1/1 sonic(config-if-Eth1/1)# no shutdown sonic(config-if-Eth1/1)# ipv6 address 2001::ad/64 sonic(config-if-Eth1/1)# end</pre> | <pre>sonic# configure sonic(config)# interface Eth1/2 sonic(config-if-Eth1/1)# no shutdown sonic(config-if-Eth1/1)# ipv6 address 2001::ac/64 sonic(config-if-Eth1/1)# end</pre> |

## Virtual routing and forwarding

Virtual Routing and Forwarding (VRF) partitions a physical router into multiple virtual routers. The control and data plane are isolated in each virtual router; traffic does not flow across virtual routers. VRF allows multiple instances of independent routing tables to co-exist within the same router simultaneously.

SONiC supports a management VRF instance, a default VRF instance, and nondefault VRF instances. Use the default and nondefault VRF instances to configure routing.

By default, the system initially assigns all physical interfaces and all logical interfaces to the default VRF instance. You can move the management interface from the default to management VRF instance. You must create the management VRF instance because it does not exist in the system by default.

In VRF configuration mode, you can configure various features for a specific VRF instance. This table shows the features and services that are supported in the management, default, and nondefault VRF instances.

**Table 34. Supported VRF features**

| Feature    | Management VRF | Default VRF | Nondefault VRF |
|------------|----------------|-------------|----------------|
| BGP        | No             | Yes         | Yes            |
| ICMP/ping  | Yes            | Yes         | Yes            |
| NTP client | Yes            | Yes         | Yes            |
| NTP server | Yes            | Yes         | No             |
| SCP client | Yes            | Yes         | Yes            |
| Traceroute | Yes            | Yes         | Yes            |

## Management VRF

The default VRF routing table is used by default on all data-plane switch ports. Although the management interface is assigned by default to the default VRF instance, you can move it to its own VRF instance.

### Create management VRF

The management VRF provides a separate `mgmt` routing table for an out-of-band management network that is different from the default in-band data-plane network. A dedicated management VRF provides an isolated routing table for the management interface when accessing the switch using SSH. The management VRF allows you to create a separate IP network that is "always on" for switch management.

```
sonic(config)# ip vrf mgmt
```

- NTP uses the management VRF by default; if the management VRF is not configured, NTP uses the default VRF.
- The management VRF does not support IP services, such as TFTP and FTP. DNS is supported.
- Use `no ip vrf mgmt` to delete the management VRF.
- SONiC does not support the use of a static IP or IPv6 route to direct incoming traffic in a remote management session, such as SSH, to the management interface.

### View management VRF

```
sonic# show ip vrf mgmt
VRF-Name Interfaces

mgmt Management0
```

## Configure nondefault VRF instances

Besides a management VRF and the default VRF instance, you can configure multiple instances of routing and forwarding tables on the switch. Multiple VRF instances are necessary when you must maintain separate traffic for different customers or different departments in a large enterprise.

When you create a nondefault VRF, you assign L3 interfaces to the VRF. This is called binding a L3 interface to a VRF. You can bind physical Ethernet, loopback, port channel, and VLAN interfaces to a nondefault VRF.

### Create VRF

```
sonic(config)# ip vrf Vrf_red
sonic(conf-vrf) #
```

- `vrf-name` is up to 15 alphanumeric characters and must start with `Vrf`; for example, `Vrf10` or `Vrf_sales`.
- Use `no ip vrf vrf-name` to delete a VRF.

### Bind L3 interface to VRF

```
sonic(config)# interface Eth1/4
sonic(conf-if-Eth1/4)# ip vrf forwarding Vrf_red
sonic(conf-if-Eth1/4)#
sonic(conf-if-Eth1/4)#
sonic(config)# exit
sonic(config)# interface Eth1/5
sonic(conf-if-Eth1/5)# ip vrf forwarding Vrf_1
%Error: No instance found for 'Vrf_1'
```

### Unbind L3 interface to VRF

```
sonic(config)# interface Eth1/25
sonic(conf-if-Eth1/25)# no ip vrf forwarding Vrf_red
Success
```

### View VRF configuration

```
sonic# show ip vrf
VRF-NAME INTERFACES

mgmt Management0
Vrf_blue Eth1/3
 Loopback20
 PortChannel120
 Vlan20
Vrf_red Eth1/2
 Loopback10
 PortChannel110
 Vlan10
```

```
sonic# show ip vrf mgmt
VRF-Name Interfaces

mgmt Management0
```

```
sonic# show ip vrf Vrf_blue
VRF-NAME INTERFACES

Vrf_blue Eth1/3
 Loopback20
 PortChannel120
 Vlan20
sonic# show ip vrf Vrf_red
VRF-NAME INTERFACES

Vrf_red Eth1/2
 Loopback10
 PortChannel10
 Vlan10
```

## Border Gateway Protocol

The Border Gateway Protocol (BGP) transmits interdomain routing information within and between autonomous systems (AS). Each AS can use its own routing policy. BGP exchanges network reachability information and routing updates between peer devices.

BGP's best-path algorithm determines the best paths to a neighbor device. The use of multiple paths from one router to another adds reliability to network connections. BGP uses TCP as its transport protocol.

BGP peers exchange routes in their BGP routing tables and continue to send each other routing updates. BGP avoids creating loops between routing domains by rejecting path updates that contain the local AS.

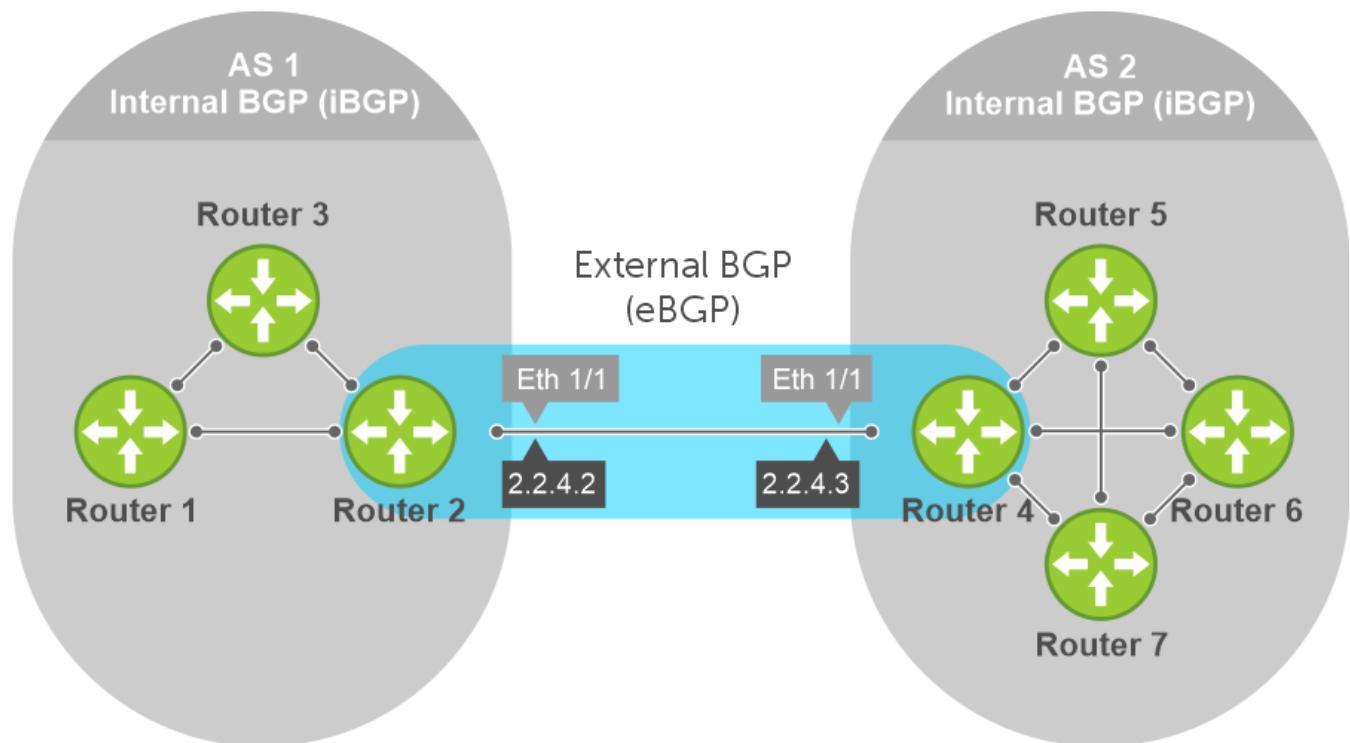
### Autonomous systems

BGP autonomous systems are a collection of nodes in individual routing domains which use local routing policies. In an AS, multiple routing protocols are supported to exchange routing updates with BGP. Each AS has a number, which an Internet authority assigns — you do not assign the BGP number.

The Internet assigned numbers authority (IANA) identifies each network with a unique AS number (ASN). AS numbers 64512 through 65534 are reserved for private purposes. AS numbers 0 and 4294967245 cannot be used in a live environment. IANA assigns valid AS numbers in the range of 1 to 64511.

|                      |                                                                                                                                                                                                                                                                                                                                         |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Multihomed AS</b> | Maintains connections to more than one other AS. This group allows the AS to remain connected to the Internet if a complete failure occurs to one of their connections. This type of AS does not allow traffic from one AS to pass through on its way to another AS.                                                                    |
| <b>Stub AS</b>       | Connected to only one AS.                                                                                                                                                                                                                                                                                                               |
| <b>Transit AS</b>    | Provides connections through itself to separate networks. For example, router 1 uses Router 2 — the transit AS, to connect to router 4. Internet service providers (ISPs) are always a transit AS because they provide connections from one network to another. An ISP uses a transit AS to sell transit service to a customer network. |

When BGP operates inside an AS - AS1 **or** AS2, it functions as an internal border gateway protocol (iBGP). When BGP operates between AS endpoints — AS1 **and** AS2, it functions as an external border gateway protocol (eBGP). iBGP provides routers inside the AS with the path to reach a router external to the AS. eBGP routers exchange information with other eBGP routers and iBGP routers to maintain connectivity and accessibility.



## Enable BGP

BGP is disabled by default. To enable BGP routing with peer devices, you must enable BGP and have at least one BGP neighbor configured on the router. The BGP neighbor can be statically or dynamically configured.

A BGP router does not automatically discover other BGP devices. You must manually configure them. To establish BGP sessions and route traffic, configure at least one BGP neighbor or peer. In BGP, routers with established TCP connections are called *neighbors* or *peers*. After a connection establishes, the neighbors exchange full BGP routing tables with incremental updates afterward. Neighbors also exchange the keepalive messages to maintain the connection.

**NOTE:** Neighbors and peers both mean the same thing — BGP-enabled routers that are directly connected or remotely connected over other BGP routers.

You can classify BGP neighbor routers and peers as internal or external. Connect eBGP peers directly, unless you enable eBGP multihop — iBGP peers do not need direct connection. The IP address of an eBGP neighbor is usually the IP address of the

interface that is directly connected to the router. The BGP process first determines if all internal BGP peers are reachable, then it determines which peers outside the AS are reachable.

These steps are required configuration settings. Repeat the configuration steps on each peer router to enable BGP route exchange between the routers.

1. Assign an autonomous system, and enter BGP configuration mode. Enter an AS number to use in the BGP routing process. Only one AS number is supported on a router.

```
sonic(config)# router bgp local_asn [vrf vrf-name]
```

- For *local\_asn*, enter a local AS number (1 to 4294967295).
- (Optional) Enter a VRF name to configure BGP in a VRF instance.

2. Configure a unique router ID using an IPv4 address. The BGP router ID is used to identify BGP peers in routing sessions.

```
sonic(conf-router-bgp)# router-id ipv4-address
```

- By default, BGP sets the router ID to the highest IPv4 address, except for the management IP address, available in the system.
- When you configure a router ID, all active BGP peer sessions are reset.

3. Configure each BGP neighbor by entering an IPv4 or IPv6 address or the local interface that is connected to the peer.

```
sonic(conf-router-bgp)# neighbor {ip-address | ipv6-address | interface {Eth slot/port[/breakout-port] | PortChannel number | Vlan vlan-id}}
```

4. Configure the AS number of the peer. Enter *internal* to configure a peer in the local AS to exchange routing information through internal BGP (iBGP) peer sessions. Enter *external* to configure a peer in an external or remote AS to exchange routing information through external BGP (eBGP) peer sessions.

```
sonic(conf-router-bgp-neighbor)# remote-as {peer_asn | internal | external}
```

5. (Optional) Enter a text description of the neighbor.

```
sonic(conf-router-bgp-neighbor)# description text
```

6. Enable BGP routing with the configured peer.

```
sonic(conf-router-bgp-neighbor)# no shutdown
```

7. Activate IPv4 or IPv6 address family.

```
sonic(conf-router-bgp-neighbor)# address-family ipv4 unicast
sonic(conf-router-bgp-neighbor-af)# activate
```

8. Repeat the steps on the peer to activate a BGP exchange with the local router.

### Enable BGP example

```
sonic(config)# router bgp 100
sonic(conf-router-bgp)# router-id 11.1.1.1
sonic(conf-router-bgp)# neighbor 5.1.1.1
sonic(conf-router-bgp-neighbor)# remote-as 1
sonic(conf-router-bgp-neighbor)# description n1_abcd
sonic(conf-router-bgp-neighbor)# no shutdown
sonic(conf-router-bgp-neighbor)# address-family ipv4 unicast
sonic(conf-router-bgp-neighbor-af)# activate
```

### View BGP routes

```
sonic# show bgp {ipv4 unicast | ipv6 unicast} [vrf vrf-name] [summary]
```

```
sonic# show bgp ipv4 unicast summary
BGP router identifier 1.1.1.1, local AS number 100
Neighbor V AS MsgRcvd MsgSent InQ OutQ Up/Down State/PfxRcd
101.2.2.2 4 200 18480 18486 0 0 1w5d19h 8
102.3.3.2 4 200 18510 18502 0 0 02:49:01 11
```

**(i) NOTE:** The show bgp ipv4 unicast summary command only displays neighbors which have the BGP address-family mode enabled to exchange IPv4 unicast routes (see [Configure BGP address family](#)).

```
sonic# show bgp ipv4 unicast
BGP routing table information for VRF default
Router identifier 2.2.2.2, local AS number 100
Status codes: R - removed, S - stale, s - suppressed, * - valid, h - history,
d - damped, > - best, = - multipath, q - queued, r - RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Path
* 10.0.0.0/8 101.2.2.2 0 100 ?
*-> 0.0.0.0 0 0 ?
*> 21.0.0.0/8 101.2.2.2 0 100 i
*> 22.0.0.0/8 101.2.2.2 0 100 i
*> 31.0.0.0/8 101.2.2.2 0 100 i
*> 32.0.0.0/8 101.2.2.2 0 100 i
*> 34.0.0.0/8 101.2.2.2 0 100 i
* 101.2.2.0/24 101.2.2.2 0 100 ?
*> 0.0.0.0 0 0 ?
```

```
sonic# show bgp ipv6 unicast
BGP routing table information for VRF default
Router identifier 2.2.2.2, local AS number 100
Status codes: R - removed, S - stale, s - suppressed, * - valid, h - history,
d - damped, > - best, = - multipath, q - queued, r - RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Path
*> 2121::/64 fe80::92b1:1cff:fef4:ab9b 0 100 i
*> 2122::/64 fe80::92b1:1cff:fef4:ab9b 0 100 i
*> 2123:3322::/64 fe80::92b1:1cff:fef4:ab9b 0 100 i
```

**(i) NOTE:** The show bgp ipv6 unicast summary command only displays neighbors which have the BGP address-family mode enabled to exchange IPv6 unicast routes (see [Configure BGP address family](#)).

```
sonic# show bgp ipv6 unicast summary
BGP router identifier 2.2.2.2, local AS number 100
Neighbor V AS MsgRcvd MsgSent InQ OutQ Up/Down State/PfxRcd
101.2.2.2 4 100 37 38 0 0 00:20:14 10
```

## Configure BGP router

After you enable BGP and enable BGP routing with peer devices, configure additional BGP features or reconfigure default settings.

### BGP defaults

BGP is enabled with these default settings:

- IPv4 unicast — Disabled
- Neighbor state-change logs — Disabled
- Fast external failover — Enabled
- Graceful restart — Disabled
- Local preference — 100
- 4-byte AS — Enabled
- Multiexit discriminator (MED) — 0
- Route flap dampening — Disabled
  - half-life — 15 minutes
  - max-suppress-time — 60 minutes
  - reuse — 750
  - suppress — 2000
- Timers:
  - keepalive — 60 seconds
  - holdtime — 180 seconds
- Add-path — Disabled

## Set BGP timers

(Optional) Configure the time (in seconds) between sending keepalive messages to a BGP neighbor (1 to 65535; default 60). Configure the hold-time to wait (in seconds) to receive a keepalive message before considering a BGP peer to be dead (3 to 65535; default 180).

```
sonic(conf-router-bgp)# timers keepalive holdtime
```

## Reset fast failover

(Optional) By default, a BGP session with an external neighbor is automatically reset if the link goes down. If you disable fast failover with the `no fast-external-failover` command, you can re-enable it.

```
sonic(conf-router-bgp)# fast-external-failover
```

- Configure BGP fast failover settings in a VRF instance from BGP VRF configuration mode.

```
sonic(config)# router bgp local_asn vrf vrf-name
sonic(conf-router-bgp-vrf)# fast-external-failover
```

## Log neighbor changes

(Optional) By default, changes in neighbor status — up and down — are automatically logged for troubleshooting purposes. To view the log, use the `bgp config` command. If you disable this log, you can reenable it with this command.

```
sonic(conf-router-bgp)# log-neighbor-changes
```

- In a VRF instance, reenable neighbor change logging from BGP VRF configuration mode:

```
sonic(config)# router bgp local_asn vrf vrf-name
sonic(conf-router-bgp)# log-neighbor-changes
```

## Best-path selection

(Optional) By default, the BGP algorithm selects the best path to a destination when given multiple alternative paths, using the prefix and path information stored in the BGP routing table. You can reconfigure the default settings that determine the best-path selection, including:

- `as-path confed` — Selects a best path that is based on AS paths in the confederation
- `as-path ignore` — Does not take into account the AS path length. The default is to use the AS path in best-path calculation.
- `as-path multipath-relax` — Permits paths of equal length to be selected to allow for load sharing. The default is to select a best path that is an exact match from multiple alternative paths.
- `as-path med {confed | missing-as-worst}` — Uses the multiexit discriminator (MED) value to select the best path learned from confederation peers if no external AS is in the path, or assigns an infinite MED value to routes that are missing the MED attribute, causing them to be the worst path alternative. The default is to assign 0 to routes with a missing MED value so that they are considered as the best alternative route.

```
sonic(conf-router-bgp)# bestpath {as-path {confed | ignore | multipath-relax [as-set] | med {confed | missing-as-worst}}}
```

## Graceful restart

(Optional) The graceful restart capability allows BGP peers to avoid a routing flap when a peer restarts. Both peers must have graceful restart that is enabled. A BGP peer maintains the routes learned from its neighbor and reestablishes the session by forwarding traffic again so that no traffic loss occurs. By default, graceful restart is disabled and a traffic loss occurs until the peer restarts. When you enable graceful restart, you can reconfigure the default restart timers:

- `restart-time seconds` — Sets the maximum time to wait for a restarting peer to resume normal operation (1 to 3600 seconds; default 120).
- `stalepath-time seconds` — Sets the maximum time to hold the paths learned from a restarting peer before deleting them (1 to 3600 seconds; default 360).
- `preserve-fw-state` — Preserves the fw state on the router after a graceful restart is performed.

```
sonic(conf-router-bgp)# graceful-restart [restart-time seconds | stalepath-time seconds | preserve-fw-state]
```

## **Coalesce timer**

(Optional) Configure the coalesce timer (in milliseconds) used in the sub-AS group to which the router belongs (1 to 4294967295).

**i | NOTE:** The coalesce timer setting has a large impact on initial convergence time. Any changes should be accompanied by careful performance testing at all scales. The intent is to allow transient changes in peer state (primarily session establishment) to settle, so that more peers can be grouped and benefit from sharing advertisement computations with the subgroup.

```
sonic(conf-router-bgp) # coalesce-time milliseconds
```

## **Read and write quanta**

**i | NOTE:** Dell Technologies recommends that you do not change the default Read and Write quanta settings.

(Optional) Configure the maximum number of packets that can be sent (`write-quanta`) or received (`read-quanta`) in a BGP peer socket during a transmission (1 to 10).

```
sonic(conf-router-bgp) # read-quanta packets
sonic(conf-router-bgp) # write-quanta packets
```

## **Route reflection**

(Optional) You can enable the sharing of route information between members of a peer group that is configured as a BGP route reflector client. Route information that is received from one peer-group member is sent to all other members.

```
sonic(conf-router-bgp) # client-to-client reflection
```

The peer-group is configured as a route reflector and its members/clients form a cluster. In a cluster, there may be multiple route reflectors for redundancy. Configure the cluster ID of the reflector that is accessed by the local router by entering its IP address in dotted-decimal format: *A.B.C.D*.

```
sonic(conf-router-bgp) # cluster-id ip-address
```

If you configure the router as a route reflector, you can enable route policies to be applied on the outbound routes that are advertised in BGP updates to neighbor clients.

```
sonic(conf-router-bgp) # route-reflector allow-outbound-policy
```

## **Deterministic MED**

(Optional) You can require that the MED value in paths that are received from the same AS is used to select the best path. The path with the lowest MED is preferred. The default is to not require the comparison of MED attribute values for best-path selection from the paths received from an AS.

```
sonic(conf-router-bgp) # deterministic med
```

## **Maximum MED at startup**

(Optional) Configure the MED value to send in BGP packets when the router starts up. The MED value determines the path choice to an AS with multiple entry points. A lower MED value is preferred in route selection to a higher value. You can enter the time (in seconds) to use the configured MED (5 to 86400). Enter the maximum MED value to use (0 to 4294967295).

```
sonic(conf-router-bgp) # max-med on-startup {time | max-med-value}
```

## **Disable eBGP route check**

(Optional) You can disable the process that verifies the required single-hop connection with an eBGP peer. This eBGP connection verification is enabled, by default, when you configure the `ebgp-mtu-hops` value as 1 for eBGP neighbors.

```
sonic(conf-router-bgp) # disable-ebgp-connected-route-check
```

## **Graceful shutdown**

(Optional) Enable the graceful shutdown feature on a BGP router to avoid traffic loss when links in iBGP and eBGP sessions are shut down for maintenance.

**i** **NOTE:** The graceful shutdown feature represents the well-known community value GRACEFUL\_SHUTDOWN 0xFFFF0000 65535:0. RFC 8326 implements Graceful BGP Session Shutdown to reduce the amount of lost traffic when BGP sessions are taken down for maintenance. The community must be supported on the peer's side to have an effect.

```
sonic(conf-router-bgp) # graceful-shutdown
```

### BGP dynamic neighbors

(Optional) Enable the dynamic neighbor process to allow BGP connections with a group of remote neighbors identified with a subnet range of IP addresses. When a BGP session starts with an IP address in the configured range, the remote peer is dynamically configured as a new member of the peer group, called a *listen range group*. The new remote peer inherits the BGP configuration of the group. No further BGP configuration is necessary. After you configure the IP address range, you can reenter the command to configure the maximum number of listen-group members allowed in the subnet.

```
sonic(conf-router-bgp) # listen [limit max-number | range ip-address/prefix-length peer-group peer-group-name]
```

### Set BGP session defaults

(Optional) Configure the default settings used in BGP peer sessions when a session is established, including:

- `ipv4-unicast` — Resets the automatic exchange of IPv4 address prefixes as the default when a BGP peer session starts. The default is to advertise only IPv4 prefix routes if you previously changed the default by entering the `no default ipv4-unicast` command.
- `local-preference value` — Sets the default local preference value (0 to 4294967295; default 100). During BGP best-path selection, the local preference value is applied to the routes exchanged with a neighbor.
- `show-hostname` — Displays the hostname in addition to the IP address of the local BGP router in show output.
- `shutdown` — Shuts down (disables) an active BGP peer session.
- `subgroup-pkt-queue-max value` — Sets the maximum number of packets that are allowed in a retransmit queue in a sub-AS.

```
sonic(conf-router-bgp) # default {ipv4-unicast | local-preference value | show-hostname | shutdown | subgroup-pkt-queue-max value}
```

### Import routes

(Optional) Checks if a BGP network route exists in an interior gateway protocol (IGP) table before importing it into the BGP routing table.

```
sonic(conf-router-bgp) # network import-check
```

### Route-map filters

(Optional) You can apply a route map to filter the exchange of incoming and outgoing BGP IPv4 or IPv6 routes. Configure the time interval (in seconds) to wait before processing received filtered routes in the BGP routing table (0 to 600; no default).

```
sonic(conf-router-bgp) # route-map delay-timer seconds
```

### Route-update delay

(Optional) You can set the time delay (in seconds) before sending routes for best-path selection and routing updates to BGP neighbors. The best-path delay is from 0 to 3600; default 120. The route-update delay is from 1 to 3600.

```
sonic(conf-router-bgp) # update-delay seconds [best-path] [route-updates]
```

### Always compare MED

(Optional) Compare the MED value received from different AS BGP neighbors during best-path selection.

```
sonic(conf-router-bgp) # always-compare-med
```

# Configure BGP address family

Use BGP address-family configuration mode to configure global address settings used to exchange IPv4 and IPv6 unicast routes, and L2VPN EVPN routes with a BGP neighbor. You can configure these address-family settings:

- Route redistribution policies
- Filter for downloading BGP routing table to RIB routes
- Multiple-path packet forwarding for iBGP and eBGP routes
- Backdoor routes for BGP address prefix
- Aggregate addresses
- Administrative distance
- Import BGP routes from a VRF

## BGP address-family configuration mode

1. Enter BGP configuration mode by entering the local AS number (1 to 4294967295).

```
sonic(config)# router bgp 100
```

2. Enter BGP address-family mode to configure IPv4, IPv6, or EVPN routes.

```
sonic(conf-router-bgp)# address-family {ipv4 unicast | ipv6 unicast | l2vpn evpn}
sonic(conf-router-bgp-af) #
```

## Configure route redistribution

Configure the redistribution of BGP-learned routes — IPv4, IPv6, or L2VPN — into another routing domain table for a different L3 protocol. Redistribute routes that are statically configured or learned through BGP connections. You can apply a route-map to redistribute only the routes that match an entry in the route map.

```
sonic(conf-router-bgp-af)# redistribute {static | connected | ospf} [route-map route-map-name]
```

## Filter BGP routes

Use a table map to filter the BGP routes downloaded as updates to the routing table. The downloaded BGP routes must match an entry in the specified route map.

```
sonic(conf-router-bgp-af)# table-map route-map-name
```

## Multipath load sharing

Configure the maximum number of eBGP and iBGP routes that can be used to forward packets over BGP using multiple paths to a neighbor:

- eBGP sessions — 1 to 256; default 1.
- iBGP sessions — 1 to 256; default 1.

```
sonic(conf-router-bgp)# maximum-paths {ibgp number | ebgp number [equal-cluster-length]}
```

## Backdoor routes

Advertise a specified network in BGP routing updates. Configure a backdoor network prefix that is learned from BGP neighbors. The backdoor prefix provides additional routing information for iBGP sessions. Configure a route map to filter the advertised networks in BGP updates. The advertised networks must match an entry in the route map.

```
sonic(conf-router-bgp-af)# network ip-prefix [backdoor] [route-map map-name]
```

## Aggregate addresses

Configure an aggregate address entry in the BGP routing table. Aggregate entries reduce the size of the routing table. An aggregate prefix combines contiguous networks into one summarized set of IP addresses. Enter *ip-prefix* in the format *ip-address/mask*.

- Use the *as-set* option to advertise the aggregate routes contained in the summary aggregate-prefix entry.
- Use the *summary-only* option to suppress the advertisement of specific routes in the prefix range to neighbors.

```
sonic(conf-router-bgp-af)# aggregate-address ip-prefix [as-set] [summary-only]
```

## Configure administrative distance

Routers use administrative distance to determine the best path between two or more routes to reach the same destination. Administrative distance indicates the reliability of the route; the lower the administrative distance, the more reliable the route. If the routing table receives route updates from one or more routing protocols for a single destination, it chooses the best route based on the administrative distance.

You can assign an administrative distance for external (eBGP), internal (iBGP), and local BGP routes:

- External (eBGP) — 1 to 255; default 20
- Internal (iBGP) — 1 to 255; default 200
- Local BGP routes — 1 to 255; default 200

```
sonic(conf-router-bgp-af)# distance bgp external-distance internal-distance local-distance
```

```
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# distance bgp 50 100 10
sonic(config-router-bgp-af)# exit
sonic(config-router-bgp)# address-family ipv6 unicast
sonic(config-router-bgp-af)# distance bgp 50 100 10
sonic(config-router-bgp-af)# exit
```

## Route dampening

 **NOTE:** Route dampening is only supported on the IPv4 unicast address family.

(Optional) You can enable BGP route dampening and reconfigure the default settings. Route dampening prevents the advertising of routes that continue to flap (go up and down). Each time a route flaps, it is assigned a penalty value of 1000, which is added each time the route flaps. By default, BGP route dampening is disabled. When enabled, these default values are used:

- *half-life-time* — Sets the time (in minutes) to wait before decreasing the accumulated penalty for a route that no longer flaps (1 to 45 minutes; default 15).
- *reuse-time* *value* — Sets the penalty value that is used to unsuppress a route when it falls below the configured value (1 to 20000; default 750).
- *suppress-route-time* — Sets the maximum time (in minutes) for a flapping route to be suppressed (1 to 255 minutes; default 60).
- *suppress-stable-route-time* — Sets the maximum time (in minutes) for a stable route to be suppressed (1 to 255 minutes; default 60).

```
sonic(conf-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# dampening [half-life-time reuse-time suppress-route-time suppress-stable-route-time]
```

## Import BGP routes from a different BGP VRF instance

To import BGP routes from a different VRF into a BGP instance, use the `import vrf vrf-name[,vrf-name] [,vrf-name] ...` command. Separate VRF names with a comma. Re-enter the command to append VRFs to the import list. For example, to import BGP routes from VRF2, VRF3, and VRF4 into VRF1:

```
sonic(config)# router bgp 100 vrf Vrf1
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# import vrf Vrf2,Vrf3
sonic(config-router-bgp-af)# import vrf Vrf4
sonic(config-router-bgp-af)# show configuration
!
address-family ipv4 unicast
import vrf Vrf2,Vrf3,Vrf4
...
```

To remove a VRF from the import list, enter the `no import vrf vrf-name` command; for example:

```
sonic(config-router-bgp-af)# no import vrf Vrf3
sonic(config-router-bgp-af)# show configuration
!
address-family ipv4 unicast
```

```
import vrf Vrf2,Vrf4
...
```

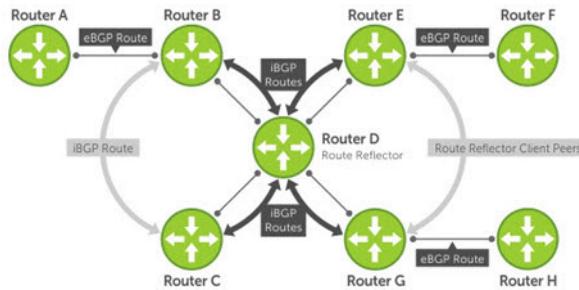
## Route reflection

Route reflectors (RRs) reorganize the iBGP core into a hierarchy and allow route advertisement rules. Route reflection divides iBGP peers into two groups — client peers and nonclient peers.

- If a route is received from a nonclient peer, it reflects the route to all client peers
- If a route is received from a client peer, it reflects the route to all nonclient and client peers

An RR and its client peers form a *route reflection cluster*. BGP speakers announce only the best route for a given prefix. RR rules apply after the router makes its best-path decision.

**(i) NOTE:** Do not use RRs in forwarding paths — hierarchical RRs that maintain forwarding plane RRs could create route loops.



Routers B, C, D, E, and G are members of the same AS — AS100. These routers are also in the same route reflection cluster, where Router D is the route reflector. Routers E and G are client peers of Router D, and Routers B and C and nonclient peers of Router D.

1. Router B receives an advertisement from Router A through eBGP. Because the route is learned through eBGP, Router B advertises it to all its iBGP peers — Routers C and D.
2. Router C receives the but does not advertise it to any peer because its only other peer is Router D (an iBGP peer) and Router D has it through iBGP from Router B.
3. Router D does not advertise the route to Router C because Router C is a nonclient peer. The route advertisement came from Router B which is also a nonclient peer.
4. Router D does reflect the advertisement to Routers E and G because they are client peers of Router D.
5. Routers E and G advertise this iBGP learned route to their eBGP peers — Routers F and H.

## Route reflector clusters

BGP route reflectors are intended for ASs with a large mesh. They reduce the amount of BGP control traffic. With route reflection configured properly, IBGP routers are not fully meshed within a cluster but all receive routing information.

You can configure clusters of routers where one router is a concentration router and the others are clients who receive their updates from the concentration router.

1. Enter router BGP mode.

```
sonic(config) # router bgp as-number
```

2. Assign an ID to a router reflector cluster; you can have multiple clusters in an AS.

```
sonic(conf-router-bgp-102) # cluster-id cluster-id
```

3. Assign a neighbor to the router reflector cluster.

```
sonic(conf-router-bgp-102) # neighbor {ip-address}
```

4. Enter Address Family mode and configure the neighbor as a route-reflector client, then return to ROUTER-BGP mode.

```
sonic(conf-router-bgp-neighbor) # address-family ipv4 unicast
sonic(conf-router-bgp-neighbor-af) # route-reflector-client
```

```
sonic(conf-router-bgp-neighbor-af) # exit
sonic(conf-router-bgp-neighbor) # exit
```

5. Assign a peer-group template as part of the router-reflector cluster.

```
sonic(conf-router-bgp) # peer-group template
```

When you enable a route reflector, the system automatically enables route reflection to all clients. To disable route reflection between all clients in this reflector, use no `client-to-client reflection` in ROUTER-BGP mode. You must fully mesh all the clients before you disable route reflection.

## Configure a route reflector for BGP

All neighbors configured with `route-reflector-client` are members of the client group, and the remaining iBGP peers are members of the nonclient group for the local route reflector.

1. Enter ROUTER-BGP mode.

```
sonic(config) # router bgp as-number
```

2. Create a remote IP address.

```
sonic(conf-router-bgp) # neighbor ip-address
```

3. Add a remote AS to the specific BGP neighbor.

```
sonic(conf-router-bgp-neighbor) # remote-as as-number
```

4. Allow sessions to use the primary IP address from a specific interface as the local address when forming a session with a neighbor.

```
sonic(conf-router-bgp-neighbor) # update source ip-address
```

5. Enter IPv4 unicast BGP-ADDRESS-FAMILY mode.

```
sonic(conf-router-bgp-neighbor) # address-family ipv4 unicast
```

6. Configure the BGP neighbor as a member of the route-reflector cluster.

```
sonic(conf-router-bgp-neighbor-af) # route-reflector-client
```

## Configure BGP neighbors

BGP neighbor configuration mode settings:

- Route redistribution policies
- Filter for downloading BGP routing table to RIB routes
- Multiple-path packet forwarding for iBGP and eBGP routes
- Backdoor routes for BGP address prefix
- Aggregate addresses
- Administrative distance

1. Enter the local AS number (1 to 65535 for a 2-byte, and 1 to 4294967295 for a 4-byte).

```
sonic(config) # router bgp 100
```

2. Enter an IPv4 address or the local port interface number that is connected to each BGP neighbor.

```
sonic(conf-router-bgp) # neighbor {ip-address | ipv6-address | interface {Eth slot/
port[/breakout-port[.subport]] | Ethernet port[.subport] | PortChannel number | Vlan
vlan-id}}
```

3. Configure the AS number of the neighbor. Enter `internal` to configure a neighbor in the local AS to exchange routing information through internal BGP (iBGP) peer sessions. Enter `external` to configure a neighbor in an external or remote AS to exchange routing information through external BGP (eBGP) peer sessions.

```
sonic(conf-router-bgp-neighbor)# remote-as {peer_asn | internal | external}
```

4. (Optional) Enter a description of the neighbor.

```
sonic(conf-router-bgp-neighbor)# description text
```

A BGP session with a configured neighbor is enabled by default. To bring down the BGP neighbor link, enter the `shutdown` command.

```
sonic(conf-router-bgp-neighbor)# shutdown
```

### Assign peer group

(Optional) The neighbor inherits the BGP configuration in the specified peer group (see [BGP peer groups](#)).

```
sonic(conf-router-bgp-neighbor)# peer-group peer-group-name
```

### eBGP multihops

(Optional) In an eBGP session, configure the maximum number of hops allowed to communicate with a peer in a remote network (1 to 255; default 255).

```
sonic(conf-router-bgp-neighbor)# ebgp-multipath hop-number
```

### Neighbor timers

(Optional) Configure the time (in seconds) between sending keepalive messages to a BGP neighbor (1 to 3600; default 60). Configure the hold-time to wait (in seconds) to receive a keepalive message before considering a BGP peer to be dead (1 to 3600; default 180). Enter a connect value (in seconds) for the retry timer (1 to 65535; default 30). The timer values that you enter override the settings that are configured for an applied peer group.

```
sonic(conf-router-bgp-neighbor)# timers keepalive holdtime [connect value]
```

### Update source

(Optional) Configure the IPv4 or IPv6 source address to use in BGP sessions with the neighbor. You can specify the Ethernet, port channel, loopback, or VLAN interface to use.

```
sonic(conf-router-bgp-neighbor)# update-source {ip-address | ipv6-address | interface
{Eth slot/port[/breakout-port[.subport]] | Ethernet port[.subport] | PortChannel
number[.subport] | Loopback number | Vlan vlan-id}}
```

### Configure route advertisement interval

(Optional) Configure the time (in seconds) between sending BGP route updates to neighbors (0 to 600; Defaults: eBGP 30 seconds, iBGP 5 seconds).

```
sonic(conf-router-bgp-neighbor)# advertisement-interval seconds
```

### Enable BFD

(Optional) Enable bi-directional forwarding detection (BFD) to detect forwarding-path failures in BGP routes. Use BFD to reduce BGP convergence time if there is a link failure.

```
sonic(conf-router-bgp-neighbor)# bfd
```

### Capability

(Optional) Enable BGP to negotiate the dynamic or extended-nexthop capability with a peer. Use the `capability dynamic` command to enable dynamic BGP peering. Use the `extended-multipath` command to allow BGP to install IPv4 routes with IPv6 next-hops if you do not have IPv4 activated on an interface.

```
sonic(conf-router-bgp-neighbor)# capability {dynamic | extended-multipath}
```

## **Disable connected checking**

(Optional) Disable the automatic checking of the connected eBGP neighbor.

```
sonic(conf-router-bgp-neighbor) # disable-connected-check
```

## **Disable capability negotiation**

(Optional) Disable automatic capability negotiation with the specified neighbor.

```
sonic(conf-router-bgp-neighbor) # dont-capability-negotiate
```

## **Enforce first AS**

(Optional) Deny any route updates received from the eBGP neighbor that does not have its AS number at the beginning of the AS\_PATH attribute in the route update.

```
sonic(conf-router-bgp-neighbor) # enforce-first-as
```

## **Enforce multihop**

(Optional) Deny any route updates received from an eBGP neighbor that is directly connected.

```
sonic(conf-router-bgp-neighbor) # enforce-multipath
```

## **Advertise local AS**

(Optional) Advertise routes with the local AS number to the BGP peer if it is part of the local autonomous system.

```
sonic(conf-router-bgp-neighbor) # local-as AS-number
```

## **Override capability**

(Optional) Enable the capability to override the negotiated best path selection.

```
sonic(conf-router-bgp-neighbor) # override-capability
```

## **Configure passive transport**

(Optional) Configure a passive transport connection with the neighbor so that the local router does not initiate a session.

```
sonic(conf-router-bgp-neighbor) # passive
```

## **Configure password**

(Optional) Configure a password for MD5 authentication on the connection with the BGP neighbor. Enter a text string in plain text or encrypted format. If you enter an encrypted password, you must specify the encrypted option.

```
sonic(conf-router-bgp-neighbor) # password text [encrypted]
```

## **Configure destination TCP port**

(Optional) Set the destination TCP port to the configured value.

```
sonic(conf-router-bgp-neighbor) # port number
```

## **Configure solo**

(Optional) Configure the capability that prevents routes advertised by the specified neighbor from being reflected back to the neighbor. Use this command only if there is a single peer that is defined in the peer group.

```
sonic(conf-router-bgp-neighbor) # solo
```

## **Strict capability match**

(Optional) Configure the requirement that the configured local BGP capabilities must exactly match the configured BGP capabilities on the neighbor.

```
sonic(conf-router-bgp-neighbor) # strict-capability-match
```

## Maximum TTL security hops

(Optional) Configure the maximum number of hops between the router and the BGP neighbor (1 to 254; no default).

```
sonic(conf-router-bgp-neighbor) # ttl-security hops number
```

## BGP sessions on IPv6 links

(Optional) Configure a neighbor interface link to require that BGP sessions can only be established on the neighbor's IPv6 link local address.

```
sonic(conf-router-bgp) # neighbor interface Eth slot/port
sonic(conf-router-bgp-neighbor) # v6only
```

To remove this requirement and return to the BGP neighbor setting that you configured with the `neighbor` command, enter the `no v6only` command.

**i** **NOTE:** The `v6only` command is available only for a BGP neighbor that is configured on an interface by specifying `neighbor interface Eth slot/port[.subport]`.

## BGP neighbor address family

Use BGP neighbor address-family configuration mode to configure global settings for the IPv4 and IPv6 unicast routes, and L2VPN EVPN routes exchanged with a BGP neighbor. You can configure various neighbor address-family settings, including:

- Advertise all paths and best path.
  - Advertise ORF capabilities.
  - Originate the default route.
  - Use a filter list or prefix list for inbound and outbound route updates.
  - Configure route reflector and client.
  - Soft reconfiguration, maximum number of received prefixes, and default weight.
1. Enter the local AS number (1 to 65535 for a 2-byte, and 1 to 4294967295 for a 4-byte).

```
sonic(config) # router bgp 100
```

2. In BGP configuration mode, enter an IP address or the local port interface number that is connected to each BGP neighbor.

```
sonic(conf-router-bgp) # neighbor {ip-address | ipv6-address | interface {Eth slot/
port[/breakout-port[.subport]] | Ethernet port[.subport] | PortChannel number | Vlan
vlan-id}}
```

3. Configure the AS number of the neighbor. Enter `internal` to configure a neighbor in the local AS to exchange routing information through internal BGP (iBGP) peer sessions. Enter `external` to configure a neighbor in an external or remote AS to exchange routing information through external BGP (eBGP) peer sessions.

```
sonic(conf-router-bgp-neighbor) # remote-as {peer-asn | internal | external}
```

4. (Optional) Enter a description of the neighbor.

```
sonic(conf-router-bgp-neighbor) # description text
```

A BGP session with a configured neighbor is enabled by default. To bring down the BGP neighbor link, enter the `shutdown` command.

```
sonic(conf-router-bgp-neighbor) # shutdown
```

## Active route exchange

(Optional) Enable route advertisement for a specified address family with a BGP neighbor. To disable address-family exchange with a neighbor, enter the `no activate` command.

```
sonic(conf-router-bgp-neighbor-af) # activate
```

## Allow AS path

(Optional) Configure the AS number to be accepted in the route exchange with a BGP neighbor.

```
sonic(conf-router-bgp-neighbor-af) # allow as-in as-count [origin]
```

- Enter an *as-count* value for the number of times that an AS number is allowed in the AS path attribute in BGP route updates.
- Enter *origin* to accept only route updates that originate from the local AS.

### Advertise best path

(Optional) Enable the advertisement of only the best path to each AS for a BGP neighbor.

```
sonic(conf-router-bgp-neighbor-af) # addpath-tx-bestpath-per-AS
```

### Override route exchange

(Optional) Enable the override of outbound route updates to an AS path that includes the remote AS configured with the BGP neighbor *remote-as* command.

```
sonic(conf-router-bgp-neighbor-af) # as-override
```

### Do not change BGP attribute

(Optional) Do not change the AS-path, MED, or next-hop attribute when advertising routes to a BGP neighbor.

```
sonic(conf-router-bgp-neighbor-af) # attribute-unchanged {as-path | med | next-hop}
```

### Advertise ORF capability

(Optional) Configure an outbound route filter (ORF) capability for sending and receiving routes from a BGP neighbor. Enter the prefix list used to filter the routes. Configure the prefix list with the *prefix-list prefix-list-name {in | out | both}* command.

```
sonic(conf-router-bgp-neighbor-af) # capability orf prefix-list {send | receive | both}
```

### Originate default route

(Optional) Configure the default route 0.0.0.0 on the local router to be used as the originating route in route advertisements sent to a BGP neighbor. Configure a route map to select, using matching route entries, the routes in which to advertise the default originating route.

```
sonic(conf-router-bgp-neighbor-af) # default-originate [route-map route-map-name]
```

### Use filter list

(Optional) Configure a BGP filter list to permit or deny route entries based on an AS path list with AS path entries. Enter *in* or *out* to apply the matching permit and deny entries to inbound or outbound route advertisements.

```
sonic(conf-router-bgp-neighbor-af) # filter-list as-path-list {in | out}
```

### Use prefix list

(Optional) Configure a prefix filter list to be used on inbound and outbound route updates. Configure the prefix list using the *ip prefix-list* command. Enter *in* or *out* to apply the matching permit and deny entries in the prefix list to inbound or outbound route advertisements.

```
sonic(conf-router-bgp-neighbor-af) # prefix-list prefix-list-name {in | out}
```

### Configure next-hop self

(Optional) Configure the local router to be the next-hop for outgoing routes on a BGP neighbor. Enter *force* to set the next hop to the local router for reflected routes. This command is useful in an AS subnet in which all neighbors do not speak to each other.

```
sonic(conf-router-bgp-neighbor-af) # next-hop-self [force]
```

### Remove private AS numbers

(Optional) Remove private AS numbers from advertised AS route-paths to a BGP neighbor. Private AS numbers are from 64512 to 65535.

```
sonic(config-router-bgp-neighbor-af) # remove-private-AS [all] [replace-AS]
```

- Enter all to remove all private AS numbers in advertised AS route paths.
- Enter replace-AS to replace advertised AS numbers with the local AS.

### Configure route-reflector client

(Optional) Configure the local router as a route reflector in the AS and the BGP neighbor as a client. As a route reflector, the router passes on all BGP-learned routes to neighbor clients. Use a route reflector in iBGP sessions when all BGP routers in the AS are not fully meshed.

```
sonic(conf-router-bgp-neighbor-af) # route-reflector-client
```

### Configure route server and client

(Optional) Configure the local router as a route server in the AS and a BGP neighbor as a client. As a route server, the router sends all BGP-learned routes with a common next-hop, AS-path, and MED values to neighbor clients.

```
sonic(conf-router-bgp-neighbor-af) # route-server-client
```

### Send communities attribute

(Optional) Configure the local router to send the communities attribute value in route updates to a BGP neighbor. Specify whether to send only standard community, only extended community, both standard and extended community, large community, or all community attributes. By default, both standard and extended community attribute values are sent in route updates.

```
sonic(conf-router-bgp-neighbor-af) # send-community {standard | extended | both | large | all | none}
```

### Configure soft reconfiguration

(Optional) A soft BGP reconfiguration allows the router to start storing inbound route updates received from a router when a BGP session is established.

```
sonic(conf-router-bgp-neighbor-af) # soft-reconfiguration inbound
```

### Unsuppress routes

(Optional) Reconfigure the advertisement of previously suppressed routes that were aggregated into a single entry in the BGP routing table. Configure a route map to select the routes to unsuppress and advertise using matching route entries.

```
sonic(conf-router-bgp-neighbor-af) # unsuppress-map route-map-name
```

### Set default weight

(Optional) Configure the default weight to assign to routes learned from a BGP neighbor; 0 to 65535, default 0. When multiple routes are available to an eBGP, the route with the highest weight is used.

```
sonic(conf-router-bgp-neighbor-af) # weight value
```

### Set maximum prefix

(Optional) Configure the maximum number of prefixes that can be received from a neighbor and stored in the BGP routing table.

```
sonic(config-router-bgp-neighbor-af) # maximum-prefix maximum-number {threshold-value | warning-only | restart number}
```

- Enter a *threshold-value* to specify the percentage of the *maximum-number* when a warning message is displayed.
- Enter *warning-only* to enter a log message and not terminate the session when the *maximum-number* is reached.
- Enter a *restart number* to specify after how many received prefixes, the local router restarts the session.

### Configure BGP neighbor address family example

```
sonic(config) # router bgp 100
sonic(config-router-bgp) # router-id 2.2.2.2
sonic(config-router-bgp) # default ipv4-unicast
```

```

sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
sonic(config-router-bgp)# address-family ipv6 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit

sonic(config-router-bgp)# neighbor 101.2.2.2
sonic(config-router-bgp-neighbor)# remote-as 100
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
sonic(config-router-bgp-neighbor)# address-family ipv6 unicast
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
sonic(config-router-bgp-neighbor)# exit

sonic(config-router-bgp)# neighbor 1001:2222::2
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
sonic(config-router-bgp-neighbor)# address-family ipv6 unicast
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
sonic(config-router-bgp-neighbor)# end

```

## **View address-family settings**

```

sonic# show bgp ipv4 unicast neighbors

BGP neighbor is 101.2.2.2, remote AS 100, local AS 100, internal link
 BGP version 4, remote router ID 1.1.1.1 , local router ID 2.2.2.2
 BGP state = Established, up for 00:20:23
 Last read 00:00:22, Last write 00:00:22
 Hold time is 180 seconds, keepalive interval is 60 seconds
 Minimum time between advertisement runs is 30 seconds
 Neighbor capabilities:
 4 Byte AS: advertised and received
 AddPath: advertised and received
 Route refresh: advertised and received
 Multiprotocol Extension: advertised and received
 Message statistics:
 InQ depth is 0
 OutQ depth is 0
 Sent Rcvd
 Opens: 2 2
 Notifications: 2 0
 Updates: 8 8
 Keepalive: 22 22
 Route Refresh: 0 0
 Capability: 0 0
 Total: 34 32

 For address family: IPv4 Unicast
 Address-family enabled
 Prefixes received 2
 For address family: IPv6 Unicast
 Address-family enabled
 Prefixes received 1
 Connections established 2, dropped 1
 Last reset 00:20:24, Last reset reason BGP Notification send
 Local host: 101.2.2.1, Local port: 55388
 Foreign host: 101.2.2.2, Foreign port: 179
 BGP Connect Retry Timer in Seconds 120

BGP neighbor is 1001:2222::2, remote AS 100, local AS 100, internal link
 BGP version 4, remote router ID 1.1.1.1 , local router ID 2.2.2.2
 BGP state = Established, up for 00:03:55
 Last read 00:00:54, Last write 00:00:54
 Hold time is 180 seconds, keepalive interval is 60 seconds
 Minimum time between advertisement runs is 30 seconds
 Neighbor capabilities:
 4 Byte AS: advertised and received

```

```

AddPath: advertised and received
Route refresh: advertised and received
Multiprotocol Extension: advertised and received
Message statistics:
 InQ depth is 0
 OutQ depth is 0
 Sent Rcvd
 Opens: 3 3
 Notifications: 2 2
 Updates: 11 11
 Keepalive: 7 7
 Route Refresh: 0 0
 Capability: 0 0
 Total: 23 23

For address family: IPv4 Unicast
 Address-family enabled
 Prefixes received 2
For address family: IPv6 Unicast
 Address-family enabled
 Prefixes received 1
Connections established 3, dropped 2
Last reset 00:03:56, Last reset reason Peer closed the session
Local host: 1001:2222::1, Local port: 40018
Foreign host: 1001:2222::2, Foreign port: 179
BGP Connect Retry Timer in Seconds 120

```

```

sonic# show bgp ipv6 unicast neighbors

BGP neighbor is 101.2.2.2, remote AS 100, local AS 100, internal link
 BGP version 4, remote router ID 1.1.1.1 , local router ID 2.2.2.2
 BGP state = Established, up for 00:20:13
 Last read 00:00:12, Last write 00:00:12
 Hold time is 180 seconds, keepalive interval is 60 seconds
 Minimum time between advertisement runs is 30 seconds
Neighbor capabilities:
 4 Byte AS: advertised and received
 AddPath: advertised and received
 Route refresh: advertised and received
 Multiprotocol Extension: advertised and received
Message statistics:
 InQ depth is 0
 OutQ depth is 0
 Sent Rcvd
 Opens: 2 2
 Notifications: 2 0
 Updates: 8 8
 Keepalive: 22 22
 Route Refresh: 0 0
 Capability: 0 0
 Total: 34 32

For address family: IPv4 Unicast
 Address-family enabled
 Prefixes received 2
For address family: IPv6 Unicast
 Address-family enabled
 Prefixes received 1
Connections established 2, dropped 1
Last reset 00:20:14, Last reset reason BGP Notification send
Local host: 101.2.2.1, Local port: 55388
Foreign host: 101.2.2.2, Foreign port: 179
BGP Connect Retry Timer in Seconds 120

BGP neighbor is 1001:2222::2, remote AS 100, local AS 100, internal link

```

# BGP peer groups

Instead of manually configuring each neighbor with the same BGP settings, you can create a peer-group policy and apply it to individual neighbors. BGP neighbors can inherit the same configuration.

## Configure peer group

Use BGP peer-group mode to configure peer groups on the router:

1. Enter BGP configuration mode by entering the router's AS number.

```
sonic(config) # router bgp local_asn
```

2. Create a peer group by assigning a name and enter Peer-Group configuration mode.

```
sonic(conf-router-bgp) # peer-group peer-group-name
```

3. Configure the AS number of the peer group. Enter `internal` to configure a peer group in the local AS to exchange routing information through internal BGP (iBGP) peer sessions. Enter `external` to configure a peer group in an external or remote AS to exchange routing information through external BGP (eBGP) peer sessions.

```
sonic(conf-router-bgp-pg) # remote-as {peer_asn | internal | external}
```

4. (Optional) Enter a text description of the peer group.

```
sonic(conf-router-bgp-pg) # description text
```

5. (Optional) In an eBGP session, configure the maximum number of hops allowed to communicate with a neighbor in a remote network (1 to 255; default 1).

```
sonic(conf-router-bgp-pg) # ebgp-multipath hop-number
```

6. (Optional) Configure the time (in seconds) between sending keepalive messages to a BGP neighbor (1 to 65535; default 60). Configure the hold-time to wait (in seconds) to receive a keepalive message before considering a BGP neighbor to be dead (3 to 65535; default 180). Enter a connect value (in seconds) for the retry time (1 to 65535; default 30).

```
sonic(conf-router-bgp-pg) # timers keepalive holdtime [connect value]
```

7. (Optional) Configure the time (in seconds) between sending BGP route updates to group members (1 to 600; eBGP 30 seconds, iBGP 5 seconds).

```
sonic(conf-router-bgp-pg) # advertisement-interval seconds
```

8. Activate the peer group settings . For information about how to add BGP neighbors to the peer group using the `peer-group` command, see [Configure BGP neighbors](#). Each neighbor that you add inherits the BGP configuration for the peer group.

```
sonic(conf-router-bgp-pg) # no shutdown
```

## Update source

(Optional) Enter the router IP address or interface to configure a BGP router to receive routing updates.

```
sonic(conf-router-bgp-pg) # update-source {ip-address | ipv6-address | interface {Eth slot/port[/breakout-port] | PortChannel number | Vlan vlan-id | Loopback number}}
```

## Enable BFD

(Optional) Enable bi-directional forwarding detection (BFD) to detect forwarding-path failures in BGP routes. Use BFD to reduce BGP convergence time if there is link failure.

```
sonic(conf-router-bgp-pg) # bfd
```

## Capability

(Optional) Enable BGP to negotiate the dynamic or extended-nexthop capability in a peer group. Use the `capability dynamic` command to enable dynamic BGP peering. Use the `extended-multipath` command to allow BGP to install IPv4 routes with IPv6 next-hops if you do not have IPv4 activated on an interface.

```
sonic(conf-router-bgp-pg) # capability {dynamic | extended-nexthop}
```

### **Disable connected checking**

(Optional) Enable peering sessions between directly connected eBGP peers using loopback addresses.

```
sonic(conf-router-bgp-pg) # disable-connected-check
```

### **Disable capability negotiation**

(Optional) Disable automatic capability negotiation in the specified peer group.

```
sonic(conf-router-bgp-pg) # dont-capability-negotiate
```

### **Enforce first AS**

(Optional) Deny any route updates received from an eBGP neighbor that does not have its AS number at the beginning of the AS\_PATH attribute in the route update.

```
sonic(conf-router-bgp-pg) # enforce-first-as
```

### **Enforce multihop**

(Optional) Enforce eBGP neighbors to perform a multihop session.

```
sonic(conf-router-bgp-pg) # enforce-multihop
```

### **Advertise local AS**

(Optional) Configure an alternate BGP autonomous system when interacting with the specified the peer group.

```
sonic(conf-router-bgp-pg) # local-as as-number
```

### **Override capability**

(Optional) Enable the override of capability negotiation using the local configured capability settings. Ignore a remote peer's capability value.

```
sonic(conf-router-bgp-pg) # override-capability
```

### **Configure passive transport**

(Optional) Configure a passive transport connection with the neighbor so that the local router does not initiate a session.

```
sonic(conf-router-bgp-pg) # passive
```

### **Configure password**

(Optional) Configure a password for MD5 authentication on the connection with a BGP neighbor. Enter a text string in plain text. If you specify the `encrypted` option, the password is stored in encrypted format.

```
sonic(conf-router-bgp-pg) # password text [encrypted]
```

### **Configure destination TCP port**

(Optional) Set the destination TCP port to configured value.

```
sonic(conf-router-bgp-pg) # port number
```

### **Configure solo**

(Optional) Configure the capability that prevents routes that are advertised by the specified neighbor from being reflected back to the neighbor. Use this command only if there is a single peer that is defined in the peer group.

```
sonic(conf-router-bgp-pg) # solo
```

## **Strict capability match**

(Optional) Configure the requirement that the configured local BGP capabilities must exactly match the configured BGP capabilities on the neighbor.

```
sonic(conf-router-bgp-pg)# strict-capability-match
```

## **Maximum TTL security hops**

(Optional) Enforces the Generalized TTL Security Mechanism (GTSM) as specified in RFC 5082. Only BGP neighbors that are the specified number of hops away (1 to 254; no default) are allowed to become neighbors. This command is mutually exclusive with the `ebgp-multipath` command.

```
sonic(conf-router-bgp-pg)# ttl-security hops number
```

## **Configure BGP peer group example**

```
sonic(config)# router bgp 100
sonic(conf-router-bgp)# router-id 2.2.2.2
sonic(conf-router-bgp)# default ipv4-unicast
sonic(conf-router-bgp)# address-family ipv4 unicast
sonic(conf-router-bgp-af)# redistribute connected
sonic(conf-router-bgp-af)# exit
sonic(conf-router-bgp)# address-family ipv6 unicast
sonic(conf-router-bgp-af)# redistribute connected
sonic(conf-router-bgp-af)# exit

sonic(conf-router-bgp)# peer-group pgrp0
sonic(conf-router-bgp-pg)# remote-as 100
sonic(conf-router-bgp-pg)# no shutdown
sonic(conf-router-bgp-pg)# exit

sonic(conf-router-bgp)# neighbor 101.2.2.2
sonic(conf-router-bgp-neighbor)# peer-group pgrp0
sonic(conf-router-bgp-neighbor)# no shutdown
sonic(conf-router-bgp-neighbor)# address-family ipv4 unicast
sonic(conf-router-bgp-neighbor-af)# activate
sonic(conf-router-bgp-neighbor-af)# exit
sonic(conf-router-bgp-neighbor)# address-family ipv6 unicast
sonic(conf-router-bgp-neighbor-af)# activate
sonic(conf-router-bgp-neighbor-af)# exit
sonic(conf-router-bgp-neighbor)# exit

sonic(conf-router-bgp)# neighbor 1001:2222::2
sonic(conf-router-bgp-neighbor)# peer-group pgrp0
sonic(conf-router-bgp-neighbor)# no shutdown
sonic(conf-router-bgp-neighbor)# address-family ipv4 unicast
sonic(conf-router-bgp-neighbor-af)# activate
sonic(conf-router-bgp-neighbor-af)# exit
sonic(conf-router-bgp-neighbor)# address-family ipv6 unicast
sonic(conf-router-bgp-neighbor-af)# activate
sonic(conf-router-bgp-neighbor-af)# exit
sonic(conf-router-bgp-neighbor)# exit
sonic(conf-router-bgp)# end
```

## **View peer groups**

```
sonic# show bgp all peer-group

BGP peer-group pgrp0, remote AS 100
Configured address-families: IPv4 Unicast;
Peer-group members:
 1001:2222::2 Established
 101.2.2.2 Established
```

## BGP peer group address family

Use BGP peer-group address-family configuration mode to configure global settings for the IPv4 and IPv6 unicast routes, and L2VPN EVPN routes exchanged with routers in a BGP peer group. You can configure various neighbor address-family settings, including:

- Advertise all paths and best path.
- Advertise ORF capabilities.
- Originate the default route.
- Use a filter list or prefix list for inbound and outbound route updates.
- Configure a route reflector and client.
- Soft reconfiguration, maximum number of received prefixes, and default weight.

### BGP peer group address family configuration mode

1. Enter BGP configuration mode by entering the local AS number (1 to 4294967295).

```
sonic(config)# router bgp 100
```

2. Enter the peer group name, and enter Peer-Group configuration mode.

```
sonic(conf-router-bgp)# peer-group peer-group-name
```

3. Enter BGP peer-group address-family mode by specifying the address-family of routes — IPv4 unicast, IPv6 unicast, or L2VPN EVPN.

```
sonic(conf-router-bgp-pg)# address-family { ipv4 unicast | ipv6 unicast | l2vpn evpn }
sonic(conf-router-bgp-pg-af) #
```

### Activate route exchange

Enable route advertisement for a specified address family with peer-group members. To disable address-family exchange, enter the no `activate address-family` command.

```
sonic(conf-router-bgp-pg-af) # activate
```

### Allow AS path

Configure the AS number to be accepted in the route exchange in a BGP peer group.

```
sonic(conf-router-bgp-pg-af) # allow as-in AS-count [origin]
```

- Enter an *as-count* value for the number of times that an AS number is allowed in the AS path attribute in BGP route updates.
- Enter *origin* to accept only route updates that originate from the local AS.

### Advertise all paths

Enable BGP Additional Path capability and advertises multiple paths for the same prefix.

```
sonic(conf-router-bgp-pg-af) # addpath-tx-all-paths
```

### Advertise best path

Enable BGP Additional path capability, advertise multiple paths for the same prefix but allow only one path to be selected per autonomous system.

```
sonic(conf-router-bgp-pg-af) # addpath-tx-bestpath-per-AS
```

### Override route exchange

Enable the override of outbound route updates to an AS path that includes the remote AS configured in a BGP peer group `remote-as` command.

```
sonic(conf-router-bgp-pg-af) # as-override
```

### Do not change BGP attribute

Do not change the AS-path, MED, or next-hop attribute value while advertising routes to a BGP neighbor.

```
sonic(conf-router-bgp-pg-af) # attribute-unchanged {as-path | med | next-hop}
```

### Advertise ORF capability

Configure an outbound route filter (ORF) capability for sending and receiving routes in a BGP peer group. Enter the prefix list used to filter the routes. Configure the prefix list with the `prefix-list prefix-list-name {in | out | both}` command.

```
sonic(conf-router-bgp-pg-af) # capability orf prefix-list {send | receive | both}
```

### Originate default route

Use the `default-originate` command to configure a default route to a BGP neighbor. Specify a route map to advertise the default route only when a matching route is present.

```
sonic(conf-router-bgp-pg-af) # default-originate [route-map route-map-name]
```

### Use filter list

Configure a BGP filter list to permit or deny route entries based on the specified AS path access-list. Enter `in` or `out` to apply the matching permit and deny entries to inbound or outbound route advertisements.

```
sonic(conf-router-bgp-pg-af) # filter-list as-path-list {in | out}
```

### Use prefix list

Configure a prefix filter list to be used on inbound and outbound route updates. Configure the prefix list using the `ip prefix-list` command. Enter `in` or `out` to apply the matching permit and deny entries in the prefix list to inbound or outbound route advertisements.

```
sonic(conf-router-bgp-pg-af) # prefix-list prefix-list-name {in | out}
```

### Configure next-hop self

Configure the local router to be the next-hop for outgoing routes in a BGP peer group. Enter `force` to set the next hop to the local router for reflected routes. This command is useful in an AS subnet in which all peer-group members do not speak to each other.

```
sonic(conf-router-bgp-pg-af) # next-hop-self [force]
```

### Remove private AS numbers

Remove private AS numbers from advertised AS route-paths in a BGP peer group. Private AS numbers are from 64512 to 65535.

```
sonic(conf-router-bgp-pg-af) # remove-private-as [all] [replace-as]
```

- Enter `all` to remove all private AS numbers in advertised AS route paths.
- Enter `replace-as` to replace advertised AS numbers with the local AS.

### Configure route-reflector client

Configure the local router as a route reflector in the AS and the peer-group members as clients. As a route reflector, the router passes on all BGP-learned routes to peer-group clients. Use a route reflector in iBGP sessions when all BGP routers in the AS are not fully meshed.

```
sonic(conf-router-bgp-pg-af) # route-reflector-client
```

### Configure route server and client

Configure the local router as a route server in the AS and the peer-group members as clients. As a route server, the router sends all BGP-learned routes with a common next-hop, AS-path, and MED values to peer-group clients.

```
sonic(conf-router-bgp-pg-af) # route-server-client
```

### Send communities attribute

Configure the local router to send the communities attribute value in route updates to peer-group members. Specify whether to send only standard communities, only extended communities, or both. By default, both standard and extended community attribute values are sent in route updates.

```
sonic(conf-router-bgp-pg-af) # send-community {standard | extended | both}
```

### Configure soft reconfiguration

A soft BGP reconfiguration allows the router to start storing inbound route updates that are received from peer-group members when a BGP session is established.

```
sonic(conf-router-bgp-pg-af) # soft-reconfiguration inbound
```

### Unsuppress routes

Reconfigure the advertisement of previously suppressed routes that were aggregated into a single entry in the BGP routing table. Configure a route map to select, using matching route entries, the routes to unsuppress and advertise.

```
sonic(conf-router-bgp-pg-af) # unsuppress-map route-map-name
```

### Set default weight

Configure the default weight to assign to routes learned from peer-group members, from 0 to 65535; default 0. When multiple routes are available to an eBGP peer, the route with the highest weight is used.

```
sonic(conf-router-bgp-pg-af) # weight value
```

### Set maximum prefix

Configure the maximum number of prefixes that can be received from peer-group members and stored in the BGP routing table.

- Enter a *threshold-value* to specify the percentage of the *maximum-number* when a warning message is displayed.
- Enter *warning-only* to generate a log message and not terminate the session, when the *maximum-number* is reached.
- Enter a *restart number* to specify after how many received prefixes, the local router restarts the session.

```
sonic(conf-router-bgp-pg-af) # maximum-prefix maximum-number {threshold-value | warning-only | restart number}
```

### Configure BGP peer-group IPv4 address-family

```
sonic(conf-router-bgp) # peer-group pgrp0
sonic(conf-router-bgp-pg) # address-family ipv4 unicast
sonic(conf-router-bgp-pg-af) # activate
sonic(conf-router-bgp-pg) # exit
```

### View BGP peer-group address family

```
sonic# show bgp ipv4 unicast neighbors

BGP neighbor is 101.2.2.2, remote AS 100, local AS 100, internal link
 BGP version 4, remote router ID 1.1.1.1 , local router ID 2.2.2.2
 BGP state = Established, up for 00:02:34
 Last read 00:00:32, Last write 00:00:32
 Hold time is 180 seconds, keepalive interval is 60 seconds
 Minimum time between advertisement runs is 30 seconds
 Neighbor capabilities:
 4 Byte AS: advertised and received
 AddPath: advertised and received
 Route refresh: advertised and received
 Multiprotocol Extension: advertised and received
 Message statistics:
 InQ depth is 0
 OutQ depth is 0
 Sent Rcvd
 Opens: 2 2
 Notifications: 2 0
 Updates: 10 14
 Keepalive: 44 44
 Route Refresh: 0 0
 Capability: 0 0
```

```

Total: 58 60

For address family: IPv4 Unicast
Address-family enabled
 Prefixes received 3
For address family: IPv6 Unicast
Address-family enabled
 Prefixes received 2
Connections established 2, dropped 1
Last reset 00:02:34, Last reset reason BGP Notification send
Local host: 101.2.2.1, Local port: 35616
Foreign host: 101.2.2.2, Foreign port: 179
BGP Connect Retry Timer in Seconds 120
...

```

## BGP routing policy filters

When you configure connections with BGP neighbors in external networks (remote AS), create routing policies to filter the routes in inbound and outbound route advertisements.

- To filter the prefixes in route advertisements, use IP prefix lists, route maps, autonomous-system-path access lists, and filter lists. Inbound and outbound route updates are managed according to the permit and deny statements that they match.
- To filter route advertisements, you can also configure BGP routers to process the Communities attribute. Routes that belong to the same community receive common treatment.

## IP prefix lists

To configure IP prefix filtering in BGP route advertisements, create a prefix list with permit and deny statements for matching network prefixes. Enter the prefix list in BGP neighbor and BGP peer-group address family configurations to filter the inbound and outbound route advertisements using the `prefix-list list {in | out}` command.

```
sonic(config)# ip prefix-list prefix-list-name seq number {permit | deny} ip-address/
prefix-length [ge ge-min-prefix-length] [le le-max-prefix-length]
```

```
sonic(config)# ipv6 prefix-list prefix-list-name seq number {permit | deny} ipv6-address/
prefix-length [ge ge-min-prefix-length] [le le-max-prefix-length]
```

- The sequence number is a mandatory parameter. Enter a sequence number to specify the order in which IP and IPv6 prefix entries are processed (1 to 4294967295). A permit/deny entry with a lower number is processed before an entry with a higher number. When a match for an IP address prefix is found, the process stops.
- Enter `permit` or `deny` for the action to take on a matching network prefix.
- Enter an IPv4 network prefix by specifying an IP address and bit mask in the format `A.B.C.D/mask`. Enter an IPv6 address network prefix by specifying an IPv6 address and bit mask in the format `A::B/mask`.
- (Optional) Specify a range of addresses to use by entering the greater than or equal to (`ge min-prefix-length`) value or the less than or equal to (`le min-prefix-length`) value. You can also enter both values in a prefix-list entry.
  - The `ge min-prefix-length` value must be greater than or equal to the `prefix-length` value.
  - The `le min-prefix-length` value must be greater than or equal to the `ge min-prefix-length` value.

### Configure IP prefix list

```
sonic(config)# ip prefix-list testv4_list_1 seq 10 permit 1.1.1.0/24
sonic(config)# ip prefix-list testv4_list_1 seq 11 permit 2.2.2.0/24
sonic(config)# ip prefix-list testv4_list_1 seq 5 deny 1.1.1.0/24 ge 25 le 28
```

```
sonic# show ip prefix-list
IP prefix list testv4_list_1:
 seq 5 deny 1.1.1.0/24 ge 25 le 28
 seq 10 permit 1.1.1.0/24
 seq 11 permit 2.2.2.0/24
```

```
sonic(config)# no ip prefix-list testv4_list_1 seq 10 permit 1.1.1.0/24
sonic(config)# no ip prefix-list testv4_list_1 seq 5 deny 1.1.1.0/24 ge 25 le 28
```

```
sonic(config)# do show ip prefix-list
IP prefix list testv4_list_1:
 seq 11 permit 2.2.2.0/24
```

### Configure IPv6 prefix list

```
sonic(config)# ipv6 prefix-list testv6_list_1 seq 5 deny 1::/64 ge 70 le 80
sonic(config)# ipv6 prefix-list testv6_list_1 seq 11 permit 1::/64
sonic(config)# ipv6 prefix-list testv6_list_1 seq 14 permit 2::/64
```

```
sonic(config)# do show ipv6 prefix-list
IPv6 prefix list testv6_list_1:
 seq 5 deny 1::/64 ge 70 le 80
 seq 11 permit 1::/64
 seq 14 permit 2::/64
```

```
sonic(config)# no ipv6 prefix-list testv6_list_1 seq 5 deny 1::/64 ge 70 le 80
sonic(config)# no ipv6 prefix-list testv6_list_1 seq 11 permit 1::/64

sonic(config)# do show ipv6 prefix-list
IPv6 prefix list testv6_list_1:
 seq 14 permit 2::/64

sonic(config)# no ipv6 prefix-list testv6_list_1
sonic(config)# do show ipv6 prefix-list
```

## BGP communities

By default, standard and extended-communities are sent in BGP route updates. You can set customized communities for inbound and outbound routes using the `route-map set {community | extcommunity} options` commands. You can also configure community names, instead of values, for well-known communities such as `local-as`, `no-advertise`, `no-export`, and `no-peer`.

You can match received routes with community values using a BGP community-list. Configure a list of BGP community values to use in the permit/deny statements in route maps. Apply the route maps to a BGP neighbor or BGP peer-group address-family.

### Configure standard community list

```
sonic(config)# bgp community-list standard community-name [deny | permit] [community-number] [local-as] [no-advertise] [no-export] [no-peer] [any | all]
```

- Enter a *community-number* in the format `AS:NN`, where valid AS numbers are 0 to 65535 and valid network numbers are 0 to 65535.
- Enter `deny` or `permit` to specify the action to take on matching routes.
- Enter other community attributes to match in route maps, such as `local-as`, `no-advertise`, `no-export`, and `no-peer`.
- Enter `any` or `all` to specify whether at least one (`any`) or all of the configured community attributes must match for a route update to be permitted or denied. The default is `any`.

For example to configure a standard community list that denies either the `no-advertise` or `no-export` route attributes:

```
sonic(config)# bgp community-list standard comm1 deny no-advertise no-export any
```

For example, to configure a standard community list that requires that all attributes match routes from network 10 in AS 1000 and routes not received from a peer:

```
sonic(config)# bgp community-list standard comm2 permit 1000:10 no-peer
```

### Configure expanded community list

```
sonic(config)# bgp community-list expanded community-name [deny | permit] regular-expression [any | all]
```

- Enter the *regular-expression* as an ordered list. The regular expression is used to filter communities by matching it with the community attribute value.

- Enter deny or permit to specify the action to take on matching routes.
- Reenter the command to configure additional expanded community lists.

For example, in an expanded community list, match routes from networks 1 to 10 in AS 1000:

```
sonic(config)# bgp community-list expanded comm3 permit 1000:[1-10]
```

### Configure an extended community list

Use extended community lists to filter the routes in VRF instances. In an extended community, the route target (RT) determines the VRFs and neighbors that receive the routes. The site of origin (SOO) prevents routes that are learned from a neighbor from being advertised back to the originating router.

```
sonic(config)# bgp extcommunity-list standard extended-community-name [deny | permit] [rt value] [soo value] [any | all]
```

- Enter deny or permit to specify the action to take on matching routes.
- Enter the route target in `rt AA:NN` or `rt ip-address:NN` format.
- Enter the site of origin in `soo AA:NN` or `soo ip-address:NN` format.
- Reenter the command to configure additional standard extended community lists.

To configure an extended expanded community list:

```
sonic(config)# bgp extcommunity-list expanded extended-community-name [deny | permit] regular-expression [any | all]
```

- Enter deny or permit to specify the action to take on matching routes.
- Enter the `regular-expression` as an ordered list. The regular expression is used to filter communities by matching it with the community attribute value.
- Reenter the command to configure additional expanded extended community lists.

For example, in an extended standard community list, match the routes for both the route target network 10 in AS 64545, and the site-of-origin network 11 in AS 64555:

```
sonic(config)# bgp extcommunity-list extcomm2 permit rt 64545:10 all
```

For example, in an extended expanded community list, match the routes for any of the configured AA:NN expressions:

```
sonic(config)# bgp extcommunity-list expanded extcomm3 permit 102:2
sonic(config)# bgp extcommunity-list expanded extcomm3 permit 103:3
sonic(config)# bgp extcommunity-list expanded extcomm3 permit 104:4
```

## AS path lists

Create permit/deny filters for the AS paths in route advertisements using regular expressions as match criteria. Apply an AS list filter to the address families of BGP neighbors and BGP peer-groups in the inbound or outbound direction using the `filter-list list {in | out}` command.

```
sonic(config)# bgp as-path-list AS-pathlist-name [seq seq-num] {permit | deny} regular-expression
```

- Enter a text string for a BGP AS path filter list.
- (Optional) Enter a sequence number that specifies the order in which the permit/deny filter is applied to AS paths in route advertisements (1 to 4294967295). You cannot enter both sequenced and non-sequenced entries in the same AS path list.
- Enter a `regular-expression` in the format `AA:NN`. The regular expression is used to filter routes by matching the AS path in a route as an ASCII string.
- Re-enter the command to configure additional AS path lists or add matching regular expressions to an AS list.

### Configuration notes

- You cannot configure the same AS path list name twice — once with numbered entries and once with unnumbered (non-sequenced) entries. You can configure an AS path list with either all numbered or all unnumbered entries.
- In an AS path list with non-sequenced entries, you cannot enter both permit and deny filters.
- In an AS path list with sequenced entries, you can enter both permit and deny filters.

## Configure AS path list

```
sonic(config)# bgp as-path-list Aslist1 permit ^65100
sonic(config)# bgp as-path-list Aslist1 permit 65303$
```

## Route maps

Create route maps to filter routes using permit and deny statements. Configure match statements to select routes; configure set statements to modify the BGP attributes in a matching route.

Apply a route-map filter to the address families of BGP neighbors and BGP peer-groups in the inbound or outbound direction using the `route-map map-name {in | out}` command.

1. Create a route map to match the route parameters listed in the next step. Specify a permit or deny statement to configure how matching routes are handled. Enter the sequence number for the order in which the statement is processed in the map.

```
sonic(config)# route-map map-name {permit | deny} sequence-number
```

2. In route-map configuration mode, enter any of these match statements to select routes.

```
sonic(conf-route-map)# match as-path acl-name
sonic(conf-route-map)# match community community-list-name
sonic(conf-route-map)# match ext-community extcommunity-list-name
sonic(conf-route-map)# match interface interface
sonic(conf-route-map)# match ip address prefix-list prefix-list-name
sonic(conf-route-map)# match ipv6 address prefix-list prefix-list-name
sonic(conf-route-map)# match metric value
sonic(conf-route-map)# match route-type {internal | external}
sonic(conf-route-map)# match origin {egp | igr | incomplete}
sonic(conf-route-map)# match tag value
sonic(conf-route-map)# match local-preference value
sonic(conf-route-map)# match peer ip-address
sonic(conf-route-map)# match ip next-hop prefix-list prefix-list-name
sonic(conf-route-map)# call route-map-name
sonic(conf-route-map)# match source-protocol {bgp | ospf | ospf3 | static | connected}
```

3. In route-map configuration mode, enter any of these set statements to change the specified BGP attribute in matching routes.

```
sonic(conf-route-map)# set as-path prepend list
sonic(conf-route-map)# set community options
sonic(conf-route-map)# set ext-community options
sonic(conf-route-map)# set comm-list community-list-name delete
sonic(conf-route-map)# set ip next-hop number
sonic(conf-route-map)# set ipv6 next-hop number
sonic(conf-route-map)# set local-preference value
sonic(conf-route-map)# set metric value
sonic(conf-route-map)# set origin {igr | egp | incomplete}
sonic(conf-route-map)# set tag value
```

## Configure route map

```
sonic(config)# route-map map1 permit 10
sonic(conf-route-map)# match as-path ASlist1
sonic(conf-route-map)# match ext-community comm3
sonic(conf-route-map)# match interface Eth1/2
sonic(conf-route-map)# set metric 100
sonic(conf-route-map)# set origin egp
sonic(conf-route-map)# set local-preference 10000
```

To remove a configured value in a route map entry, enter the no version of the match or set command; for example:

```
sonic(config)# route-map map1 permit 10
sonic(conf-route-map)# no match as-path
sonic(conf-route-map)# no set origin
```

## Unnumbered BGP

BGP uses TCP for connections with neighbor devices. A router interface that connects to a neighbor requires a unique IP address. Assigning an IP address to each router interface may exhaust the available pool of IP addresses, resulting in an error in operation.

An unnumbered interface does not have a user-configured IP address. BGP unnumbered interfaces use the extended next-hop encoding (ENHE) feature, as defined in RFC 5549, to advertise IPv4 routes with an IPv6 next hop.

After you enable IPv6 on an interface that is connected to a BGP neighbor, an IPv6 link-local address is automatically created. BGP uses the link-local address to set up a BGP session with the neighbor. Unnumbered interfaces use IPv6 router advertisements (RAs) to identify the address of a BGP neighbor.

Using unnumbered BGP, hosts and switches automatically discover neighboring routers. Peer routers that are connected with point-to-point links are discovered by parsing their router advertisements.

Each router periodically generates an RA, which contains its MAC address and link-local address. If you configure an unnumbered interface on a BGP-enabled router, the interface parses the RA information that it receives from a peer device and sets up a BGP session with the device.

### Configure unnumbered BGP

1. Enable IPv6 on an interface:

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ipv6 enable
```

2. Configure unnumbered BGP on the interface using the `capability extended-nexthop` command.

```
sonic(config)# interface Loopback 0
sonic(conf-if-lo0)# ip address 10.0.1.1/32
sonic(conf-if-lo0)# end
sonic# configure terminal
sonic(config)# router bgp 64601
sonic(conf-router-bgp)# router-id 10.0.1.1
sonic(conf-router-bgp)# neighbor interface Eth1/1
sonic(conf-router-bgp-neighbor)# remote-as external
sonic(conf-router-bgp-neighbor)# capability extended-nexthop
sonic(conf-router-bgp-neighbor)# no shutdown
sonic(conf-router-bgp-neighbor)# address-family ipv4 unicast
sonic(conf-router-bgp-neighbor-af)# activate
sonic(conf-router-bgp-neighbor-af)# exit
sonic(conf-router-bgp-neighbor)# exit
sonic(conf-router-bgp)# end
```

## View BGP configuration and operation

You can view BGP configuration and operation using show commands.

### View BGP routing table

```
sonic# show bgp {ipv4 unicast | ipv6 unicast} [vrf vrf-name]
```

```
sonic# show bgp ipv4 unicast
BGP routing table information for VRF default
Router identifier 20.0.0.1, local AS number 100
Route status codes: * - valid, > - active, e - ECMP
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List,
LL Nexthop - Link Local Nexthop
 Network Next Hop Metric LocPref Weight Path
* > 3.0.0.1/32 1.1.1.2 0 100 0 300 i
* 3.0.0.1/32 1.0.0.2 0 100 0 200 ?
* > 3.0.0.2/32 1.1.1.2 0 100 0 300 i
* 3.0.0.2/32 1.0.0.2 0 100 0 200 ?
* > 3.0.0.3/32 1.1.1.2 0 100 0 300 i
```

## **View BGP routing table entry**

```
sonic# show bgp {ipv4 unicast | ipv6 unicast} [vrf vrf-name] ip-prefix

sonic# show bgp ipv4 unicast 30.0.0.0/24
BGP routing table entry for 30.0.0.0/24, version 35
Paths: (3 available, best #2, table default)
Multipath: eBGP
Flag: 0x860
 Advertised to update-groups:
 1
 200
 50.0.0.1 from 50.0.0.1 (20.0.0.1)
 Origin incomplete, localpref 100, valid, external, backup/repair
 Only allowed to recurse through connected route
 200
 60.0.0.1 from 60.0.0.1 (20.0.0.1)
 Origin incomplete, localpref 100, weight 100, valid, external, best
 Only allowed to recurse through connected route
 200
 70.0.0.1 from 70.0.0.1 (40.0.0.1)
 Origin incomplete, localpref 100, valid, external,
 Only allowed to recurse through connected route
```

## **View BGP neighbors**

```
sonic# show bgp {ipv4 unicast | ipv6 unicast} [vrf vrf-name] summary

sonic# show bgp ipv4 unicast summary
BGP summary information for VRF default
Router identifier 20.0.0.1, local AS number 100
Neighbor V AS MsgRcvd MsgSent InQ OutQ Up/Down State PfxRcd PfxAcc
10.1.0.100 4 200 1075 1083 0 0 00:04:04 Connect
10.2.0.101 4 200 1079 1088 0 0 00:04:14 Connect
```

## **View BGP neighbor received/advertised routes**

```
sonic# show bgp {ipv4 unicast | ipv6 unicast} [vrf vrf-name] neighbors [nbr-ip
[routes | received-routes | advertised-routes]]

sonic# show bgp ipv4 unicast neighbors 10.3.0.103 advertised-routes
BGP routing table information for VRF default
Router identifier 10.0.0.102, local AS number 64500
Route status codes: s - suppressed, * - valid, > - active, # - not installed,
 E - ECMP head, e - ECMP, S - Stale,
 c - Contributing to ECMP, b - backup, L - labeled-unicast
Origin codes: i - IGP, e - EGP, ? - incomplete
AS Path Attributes: Or-ID - Originator ID, C-LST - Cluster List,
 LL Nexthop - Link Local Nexthop

 Network Next Hop Metric LocPref Weight Path
* > 10.1.0.0/24 10.3.0.102 - 100 - i
* > 10.2.0.0/24 10.3.0.102 - 100 - i
* > 10.3.0.0/24 10.3.0.102 - 100 - i
* > 10.100.0.0/24 10.1.0.100 200 100 - 64496 i
* > 10.100.1.0/24 10.1.0.100 - 100 - 64496 64497 65536 i
* > 10.100.2.0/24 10.1.0.100 42 100 - 64496 ?
* > 10.101.0.0/24 10.2.0.101 - 100 - 64510 i
* > 10.101.1.0/24 10.2.0.101 - 100 - 64510 i
* > 10.101.2.0/24 10.2.0.101 - 100 - 64510 i
```

## **View BGP peer groups**

```
sonic# show bgp all [vrf vrf-name] peer-group [peer-group-name]
```

**i** **NOTE:** The show bgp all peer-group command displays information for all BGP peer-group address families — IPv4 unicast, IPv6 unicast, and L2VPN EVPN.

```
sonic# show bgp all peer-group
BGP peer-group is EXTERNAL
 BGP version 4
 Static peer-group members:
 VRF default:
 10.1.0.100, state: Connect
 Negotiated MP Capabilities:
 IPv4 Unicast: No
 IPv6 Unicast: No
 10.2.0.101, state: Connect
 Negotiated MP Capabilities:
 IPv4 Unicast: No
 IPv6 Unicast: No

BGP peer-group is INTERNAL
 BGP version 4
 Listen-range subnets:
 VRF default:
 10.3.0.0/24, remote AS 64500
 Dynamic peer-group members:
 VRF default:
```

#### View BGP route maps

```
sonic# show route-map
Route map map1:
 permit, sequence 10
 Match clauses:
 Set clauses:
 local preference 10
 Call clauses:
 Actions:
 Exit routemap
Route map map2:
 permit, sequence 2
 Match clauses:
 med 10
 Set clauses:
 Call clauses:
 Actions:
 Exit routemap
```

#### View BGP prefix lists

```
sonic# show ip prefix-list
IP prefix list pref1:
 seq 20 permit 20.0.0.0/8

sonic# show ipv6 prefix-list
IPv6 prefix list pref2:
 seq 1 permit 2222::/64 ge 65 le 65
 seq 2 permit 2223::/64 ge 65 le 128
```

#### View BGP community lists

```
sonic# show bgp community-list
Standard community list com1: match: ANY
 local-AS
Expanded community list com2: match: ANY
 Extended1

sonic# show bgp ext-community-list
Standard extended community list extcom1: match: ANY
 rt:2:2
```

```
Expanded extended community list extcom2: match: ANY
extcom
```

## View AS path access lists

```
sonic# show bgp as-path-access-list
AS path list aspath1:
members: 1:1+
```

## View BGP IPv4 routes

You can view the IPv4 routes in the BGP routing table using `show` commands.

### Filter the view of IPv4 routes in BGP routing table

**Syntax**      `show bgp ipv4 unicast [vrf vrf-name] {ipv4-address | ipv4-prefix/mask} [bestpath | multipath]`

**Parameters**

- `vrf vrf-name` — (Optional) Enter a VRF instance name to display of BGP IPv4 routes in the VRF. The default VRF is used by default.
- `ipv4-address` — Enter an IP address to specify a BGP neighbor or network.
- `ipv4-prefix/mask` — Enter an IP prefix or address and bit mask to display routes to only matching BGP devices in the network. The bit mask is from 1 to 32.
- `bestpath` — (Optional) Display the best path to the IP address or prefix.
- `multipath` — (Optional) Display only the routes that exactly match the community criteria.

### Examples

```
sonic# show bgp ipv4 unicast 51.0.0.3
BGP routing table entry for 51.0.0.0/24
Paths: (2 available, best #2, table default)
 200
 102.3.3.2 from 102.3.3.2 (2.2.2.2)
 Origin IGP, metric 0, valid, external, multipath
 Community: 200:555 noExport noAdvertise localAs noPeer
 Last update: 2020-05-12 18:08:38
 200
 101.2.2.2 from 101.2.2.2 (2.2.2.2)
 Origin IGP, metric 0, valid, external, multipath, best
 (Older Path)
 Community: 200:555 noExport noAdvertise localAs noPeer
 Last update: 2020-05-12 18:08:38
```

```
sonic# show bgp ipv4 unicast 51.0.0.3 bestpath
BGP routing table entry for 51.0.0.0/24
Paths: (1 available, best #1, table default)
 200
 101.2.2.2 from 101.2.2.2 (2.2.2.2)
 Origin IGP, metric 0, valid, external, multipath, best
 (Older Path)
 Community: 200:555 noExport noAdvertise localAs noPeer
 Last update: 2020-05-12 18:08:38
```

```
sonic# show bgp ipv4 unicast 51.0.0.3 multipath
BGP routing table entry for 51.0.0.0/24
Paths: (2 available, best #2, table default)
 200
 102.3.3.2 from 102.3.3.2 (2.2.2.2)
 Origin IGP, metric 0, valid, external, multipath
 Community: 200:555 noExport noAdvertise localAs noPeer
 Last update: 2020-05-12 18:08:38
 200
 101.2.2.2 from 101.2.2.2 (2.2.2.2)
 Origin IGP, metric 0, valid, external, multipath, best
 (Older Path)
 Community: 200:555 noExport noAdvertise localAs noPeer
 Last update: 2020-05-12 18:08:38
```

## View IPv4 routes in BGP communities

### Syntax

```
show bgp ipv4 unicast [vrf vrf-name] community {AA:NN | local-as | no-advertise | no-export | no-peer} [exact-match]
```

### Parameters

- *vrf vrf-name* — (Optional) Enter a VRF instance name to filter the display of BGP IPv4 routes.
- *AA:NN* — Display IPv4 routes that match the specified community number, where *AA* is an autonomous system number (1 to 65535; *NN* is a network number from 1 to 65535).
- *ipv4-prefix/mask* — Enter an IP prefix or address and bit mask (1 to 32) to display routes only to matching BGP devices in the network.
- *local-as* — Display BGP routes in the local autonomous system.
- *no-advertise* — Display BGP routes in the no-advertise community — routes not advertised to any neighbor in the local or an external autonomous system.
- *no-export* — Display BGP routes in the no-export community — routes advertised only to neighbors in the same autonomous system.
- *no-peer* — Display BGP routes in the no-peer community — routes that may not be advertised to neighbors in the local or external autonomous systems.
- *exact-match* — (Optional) Display only the BGP routes that exactly match the specified community criteria.

### Examples

```
sonic# show bgp ipv4 unicast community 200:555
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Status codes: R - removed, S - stale, s - suppressed, * - valid, h - history,
 d - damped, > - best, = - multipath, q - queued, r - RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
* 51.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 52.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
```

```
sonic# show bgp ipv4 unicast community 200:555 exact-match
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Status codes: R - removed, S - stale, s - suppressed, * - valid, h - history,
 d - damped, > - best, = - multipath, q - queued, r - RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
* 52.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
```

```
sonic# show bgp ipv4 unicast community local-as
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Status codes: R - removed, S - stale, s - suppressed, * - valid, h - history,
 d - damped, > - best, = - multipath, q - queued, r - RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
* 51.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 53.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
```

```
sonic# show bgp ipv4 unicast community no-advertise exact-match
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Status codes: R - removed, S - stale, s - suppressed, * - valid, h - history,
 d - damped, > - best, = - multipath, q - queued, r -
```

```

RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
* 56.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i

```

### Filter IPv4 routes in BGP routing table with route map

**Syntax** show bgp ipv4 unicast [vrf *vrf-name*] route-map *map-name*

**Parameters**

- *vrf vrf-name* — (Optional) Enter a VRF instance name to filter the display of BGP IPv4 routes.
- *route-map map-name* — Enter a route map to display only matching BGP routes.

**Example**

```

sonic# show bgp ipv4 unicast route-map set-next-hop-global-v6
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Status codes: R - removed, S - stale, s - suppressed, * - valid, h -
history,
d - damped, > - best, = - multipath, q - queued, r -
RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
* 21.0.0.0/8 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 51.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 52.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 53.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 54.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 55.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 56.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 57.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
*> 221.0.0.0/8 0.0.0.0 0 0 32768 i

```

### View BGP IPv4 route statistics

**Syntax** show bgp ipv4 unicast [vrf *vrf-name*] statistics

**Parameters**

- *vrf vrf-name* — (Optional) Enter a VRF instance name to filter the display of BGP IPv4 routes.

**Example**

```

sonic# show bgp ipv4 unicast statistics
BGP IPv4 Unicast RIB statistics
Total Advertisements : 17
Total Prefixes : 9
Average prefix length : 20.44
Unaggregateable prefixes : 9
Maximum aggregateable prefixes: 0
BGP Aggregate advertisements : 0
Address space advertised : 3.35562e+07
 % announced : 0.78
 /8 equivalent : 2.00
 /24 equivalent : 131079.00
Advertisements with paths : 17
Longest AS-Path (hops) : 1
Average AS-Path length (hops) : 0.94
Largest AS-Path (bytes) : 6
Average AS-Path size (bytes) : 5.65
Highest public ASN : 200

```

### View BGP IPv4 routes in VRFs

**Syntax** show bgp ipv4 unicast vrf {*vrf-name* | default | all}

## Parameters

- `vrf vrf-name` — Display the BGP IPv4 routes in the specified VRF.
- `default` — Display the BGP IPv4 routes in the default VRF.
- `all` — Display the BGP IPv4 routes in all VRFs.

## Example

```
sonic# show bgp ipv4 unicast vrf all

BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Status codes: R - removed, S - stale, s - suppressed, * - valid, h -
history,
d - damped, > - best, = - multipath, q - queued, r -
RIB-failure
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
* 56.0.0.0/24 102.3.3.2 0 0 200 i
* 21.0.0.0/8 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 51.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 52.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 53.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 54.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 55.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 56.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
* 57.0.0.0/24 102.3.3.2 0 0 200 i
*> 101.2.2.2 0 0 200 i
*> 221.0.0.0/8 0.0.0.0 0 0 32768 i

BGP routing table information for VRF Vrf_blue
Router identifier 3.3.3.3, local AS number 300
Route status codes: * - valid, > - best
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
*> 70.1.1.0/24 110.7.7.4 0 0 400 i
```

## IPv6 Neighbor Discovery Protocol

As IPv4 uses the [Address Resolution Protocol](#) (ARP) for address translation, IPv6 uses the Network Discovery Protocol (NDP) to allow host devices and routers to discover each other. NDP operates at the link layer of OSI model.

Hosts and routers use NDP to determine the link-layer addresses for neighbors known to reside on attached links. Hosts also use NDP to discover neighboring routers that can forward packets on their behalf. In addition, hosts and routers use NDP to actively track which neighbors are reachable and which are not, and to detect changes in link-layer addresses.

NDP messages are in ICMPv6 packet format. ICMPv6 is used by IPv6 hosts and routers to send error and informational messages. NDP messages use these ICMPv6 packet types:

- Router Solicitation (RS) message (ICMPv6 type 133) — To join an IPv6 network, a new host or router sends an all-router multicast request message on the local link. This message includes header information, such as ICMPv6 type (133), source link-local address, and destination address which is the all-router multicast address (FF02::2).
- Router Advertisement (RA) message (ICMPv6 type 134) — In an IPv6 network, routers advertise their presence periodically or immediately in response to a Router Solicitation message. A router advertisement uses ICMPv6 message type 134, and includes the router link-local address as the source and FF02::1 (all-node multicast address) as the destination so that every IPv6 host and router receives the message. A Router Advertisement message also includes RA data:
  - IPv6 prefix: Used to configure IPv6 addresses.
  - Address auto-configuration: Stateless or Stateful
  - Default gateway information
  - MTU
  - Timer values: router lifetime, reachable time, and retransmission time
  - Flags, such as default router preference, managed address configuration, RA retransmit interval

- Neighbor Solicitation (NS) message (ICMPv6 type 135) — Neighbor solicitations are used by hosts and routers to determine a neighbor's link-layer address or to verify that a neighbor is still reachable using a cached link-layer address. For example, if device A wants to know the MAC address of device B, it sends a Neighbor Solicitation message. The header information is in ICMPv6 type 135 format. The source MAC address is from device A; the destination MAC address is the multicast MAC address. The source address is either empty or device A's unicast IP address. The destination address is the solicited multicast address of device B. The target address field contains the destination IP address of device B, for which the MAC address is requested.
- Neighbor Advertisement (NA) message (ICMPv6 type 136) — Neighbor advertisements are used by hosts and routers to respond to a Neighbor Solicitation message. A neighbor advertisement uses ICMPv6 type 136 format. The source is the solicitation device's unicast IP address; the destination is the device's IP unicast address. The NA message packet includes the solicitation device's MAC address.
- Redirect message (ICMPv6 type 137) — A router sends a redirect message to inform a host of a better first-hop node on the path to a destination. The better first-hop can be a different router or the destination itself. For example, when device A sends packets to router B, B sends a redirect message to A which contains header information in ICMPv6 type 137 format. The source is the link-local address of router B; the destination is device A's link-local address. The target address is the IPv6 address of the better first-hop.

## Configure IPv6 router advertisement

Configure IPv6 router advertisement in Interface configuration mode on the following interfaces:

- Ethernet: sonic(conf-if-Ethslot/port) #
  - Routed VLAN: sonic(conf-if-Vlan#) #
  - Port channel: sonic(conf-if-po#) #
  - Ethernet subinterfaces: sonic(conf-if-Ethslot/port/subinterface) #
  - Port-channel subinterfaces: sonic(conf-subif-PortChannel) #
1. (Optional) Stop sending router advertisement messages for IPv6 neighbor discovery. By default, sending router advertisement messages is disabled. You can enable sending router advertisements using the `no ipv6 nd suppress-ra` command.

```
[no] ipv6 nd suppress-ra
```

2. Configure the IPv6 prefixes to be included in router advertisements for IPv6 neighbor discovery.

```
ipv6 nd prefix ipv6-prefix/prefix-length [valid-lifetime] [preferred-lifetime] [off-link] [no-autoconfig] [router-address]
```

- Enter the IPv6 prefix in hexadecimal format with a prefix-length value.
  - (Optional) Enter the lifetime (in seconds) that the IPv6 prefix is advertised as a valid address (0 to 4294967295).
  - (Optional) Enter the time (in seconds) that the IPv6 prefix is advertised as a preferred address (0 to 4294967295).
  - (Optional) Enter **offlink** to advertise the IPv6 prefix with the L-bit clear, and not add the prefix to the routing table as a Connected prefix. If the prefix was statically configured in the routing table, it is removed.
  - (Optional) Enter **no-autoconfig** to advertise the IPv6 prefix with the A-bit clear, and prevent hosts on the local link from using the prefix for IPv6 autoconfiguration.
  - (Optional) Enter **router-address** to set the R flag and communicate to hosts on the local link that the specified prefix contains a complete IPv6 address.
3. (Optional) Configure the interval used to send router advertisement messages for IPv6 neighbor discovery. Specify a time in seconds (1-1800; default 600) or milliseconds (**msec** 70-1800000).

```
ipv6 nd ra-interval {seconds | msec milliseconds}
```

4. (Optional) Configure faster transmissions of RA packets to accelerate convergence and neighbor establishment, particularly for unnumbered peering. RA fast retransmission is enabled by default. To disable it, enter the `no ipv6 nd ra-fast-retrans` command. Disabling RA fast retransmission is sometimes necessary to have IPv6 neighbor discovery compliant with the RFC by having slower convergence and neighbor establishment.

```
ipv6 nd ra-fast-retrans
```

5. (Optional) Configure the time interval (in milliseconds) for resending consecutive Advertisement Retransmit messages (0-4294967295; default 0). Setting the value to zero indicates that the RA retransmission time is not specified by the router.

```
ipv6 nd ra-retrans-interval milliseconds
```

6. (Optional) Configure the maximum number of next hops supported in router advertisements (0-255; default 0).

```
 ipv6 nd ra-hop-limit number
```

7. (Optional) Configure the time (in seconds) that the router is advertised as the default router (0-9000; default 0). Enter 0 to configure the router as a non-default router. The RA lifetime value should be greater than RA interval time.

```
 ipv6 nd ra-lifetime seconds
```

8. (Optional) Configure the time (in seconds) that an IPv6 neighbor is considered to be reachable after a reachability confirmation is received (0-3600000; default 0).

```
 ipv6 nd reachable-time seconds
```

9. (Optional) Set the "managed address configuration flag" in IPv6 router advertisements so that host devices know to use stateful autoconfiguration to receive IPv6 addresses.

```
 ipv6 nd managed-config-flag
```

10. (Optional) Set the "other stateful configuration flag" in IPv6 router advertisements so that host devices can receive autoconfiguration information besides IPv6 addresses. To receive non-address information, host devices should use stateful autoconfiguration.

```
 ipv6 nd other-config-flag
```

11. (Optional) Set the "home agent configuration flag" in IPv6 router advertisements so that the router is identified by the same IPv6 address (mobile IP) even if it moves from one network to another. When moving the router to a different network, connectivity is maintained seamlessly without user intervention.

```
 ipv6 nd home-agent-config-flag
```

12. (Optional) Set the time (in seconds) that the router can have preferred home agent status (0-65535; default 0). The value you set is applied only if the router has been configured as a home agent using the `ipv6 nd home-agent-config-flag` command.

```
 ipv6 nd home-agent-preference seconds
```

13. (Optional) Set the time (in seconds) that the router is considered as the home agent on the network (0-65535; default 0). The value you set is applied only if the router has been configured as the home agent using the `ipv6 nd home-agent-config-flag` command.

```
 ipv6 nd home-agent-lifetime seconds
```

14. (Optional) Enable the advertisement interval option in router advertisements so that an IPv6 mobile device that joins the network knows that it can receive router advertisements.

```
 ipv6 nd adv-interval-option
```

15. (Optional) Configure the default router preference sent in router advertisements: high, medium, or low (default medium). The default router preference is used by host devices to select the destination for IPv6 routing when two routers on a link provide equal next-hop routing.

```
 ipv6 nd router-preference value
```

16. (Optional) Set the maximum transmission unit (MTU) size of IPv6 messages transmitted by the router (1 to 65535; default 0). By default, no MTU size is advertised in IPv6 messages (default value 0).

```
 ipv6 nd mtu bytes
```

17. (Optional) Configure a recursive domain name server to advertise in neighbor discovery messages using the RDNSS (type 25) option described in RFC8106. Re-enter the command to configure additional recursive DNS servers. Optionally, enter the maximum time in seconds for which the specified IPv6 server address is used for domain name resolution (0-4294967295; default is three times the RA interval configured with the `ipv6 nd ra-interval` command). Enter 0 to specify that the IPv6 address is no longer used. Enter infinite to advertise a DNS server for an infinite time. By default, no recursive domain name server is advertised.

```
 ipv6 nd rdnss ipv6-address [seconds | infinite]
```

18. (Optional) Advertise the DNS search list in neighbor discovery messages, using the DNSSL (type 31) option as described in RFC8106. Enter a text string for a domain name suffix to identify DNS servers. Optionally, enter the maximum time in seconds for which the specified domain suffix is used for domain name resolution (0-4294967295; default is 3 times the RA interval configured with the `ipv6 nd ra-interval` command). Enter 0 to specify that the domain name suffix is no longer used. Enter infinite to advertise a list of DNS servers for an infinite time. Re-enter the command to configure an additional DNS server list.

```
ipv6 nd dnssl domain-name-suffix [seconds | infinite]
```

- (i) NOTE:** (Optional) The `radv enable` command is provided only for backward compatibility with community SONIC (`radv docker`) and is disabled by default. This command is not required for the switch to send router advertisements. It is strongly recommended not to enable the `radv enable` command. If you use this command to configure the switch, you must save the switch configuration and reload the switch.

```
sonic# radv enable
```

## View IPv6 router advertisement

To view IPv6 router advertisement on all interfaces:

```
sonic# show ipv6 nd ra-interfaces
Interfaces:
 Vlan100
 ND advertised reachable time is 0 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND advertised hop-count limit is 64 hops
 ND router advertisements sent: 0 rcvd: 0
 ND router advertisements are sent every 600 seconds
 ND router advertisements lifetime tracks ra-interval
 ND router advertisement default router preference is MEDIUM
 Hosts use stateless autoconfig for addresses.
 ND router advertisements with Adv. Interval option.
 Advertised Link MTU is 1200
 rdns 2001::1 1234
 rdns 2002::1 infinite
 rdns 2003::1
 dnssl mybroadcom1 1234
 dnssl mybroadcom2
 dnssl mybroadcom3 infinite
 prefix 2001::1/128 5000 4000
 prefix 2002::1/128 no-autoconfig off-link
 Vlan200
 ND advertised reachable time is 0 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND advertised hop-count limit is 64 hops
 ND router advertisements sent: 0 rcvd: 0
 ND router advertisements are sent every 600 seconds
 ND router advertisements lifetime tracks ra-interval
 ND router advertisement default router preference is MEDIUM
 Hosts use stateless autoconfig for addresses.
 Ethernet64
 ND advertised reachable time is 0 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND advertised hop-count limit is 64 hops
 ND router advertisements sent: 0 rcvd: 0
 ND router advertisements are sent every 1234 milliseconds
 ND router advertisements lifetime tracks ra-interval
 ND router advertisement default router preference is MEDIUM
 Hosts use stateless autoconfig for addresses.
 Advertised Link MTU is 900
 rdns 2001::1 infinite
 PortChannel15
 ND advertised reachable time is 0 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND advertised hop-count limit is 64 hops
 ND router advertisements sent: 0 rcvd: 0
 ND router advertisements are sent every 600 seconds
 ND router advertisements lifetime tracks ra-interval
```

```

ND router advertisement default router preference is MEDIUM
Hosts use stateless autoconfig for addresses.

 Ethernet64.12
 ND advertised reachable time is 0 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND advertised hop-count limit is 64 hops
 ND router advertisements sent: 0 rcvd: 0
 ND router advertisements are sent every 600 seconds
 ND router advertisements lifetime tracks ra-interval
 ND router advertisement default router preference is MEDIUM
 Hosts use stateless autoconfig for addresses.

 PortChannel15.20
 ND advertised reachable time is 0 milliseconds
 ND advertised retransmit interval is 0 milliseconds
 ND advertised hop-count limit is 64 hops
 ND router advertisements sent: 0 rcvd: 0
 ND router advertisements are sent every 600 seconds
 ND router advertisements lifetime tracks ra-interval
 ND router advertisement default router preference is MEDIUM
 Hosts use stateless autoconfig for addresses.

...

```

## View BGP IPv6 routes

You can view the IPv6 routes in the BGP routing table using `show` commands.

### Filter the view of IPv6 routes in BGP routing table

|               |                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | <code>show bgp ipv6 unicast [vrf vrf-name] {ipv4-address   ipv4-prefix/mask} [bestpath   multipath]</code> |
|---------------|------------------------------------------------------------------------------------------------------------|

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Parameters</b> | <ul style="list-style-type: none"> <li>• <code>vrf vrf-name</code> — (Optional) Enter a VRF instance name to display of BGP IPv6 routes in the VRF. The default VRF is used by default.</li> <li>• <code>ipv6-address</code> — Enter an IPv6 address to specify a BGP neighbor or network.</li> <li>• <code>ipv6-prefix/mask</code> — Enter an IPv6 prefix or address and bit mask (1 to 128) to display routes to only matching BGP devices in the network.</li> <li>• <code>bestpath</code> — (Optional) Display the best path to the IP address or prefix.</li> <li>• <code>multipath</code> — (Optional) Display only the routes that exactly match the community criteria.</li> </ul> |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### Examples

```

sonic# show bgp ipv6 unicast 5001:1111::22
BGP routing table entry for 5001:1111::/64
Paths: (2 available, best #2, table default)
 200
 1002:3333::2 from 1002:3333::2 (2.2.2.2)
 Origin IGP, metric 0, valid, external, multipath
 Community: 600:755 noExport noAdvertise localAs noPeer
 Last update: 2020-05-12 18:09:04
 200
 1001:2222::2 from 1001:2222::2 (2.2.2.2)
 Origin IGP, metric 0, valid, external, multipath, best
 (Older Path)
 Community: 600:755 noExport noAdvertise localAs noPeer
 Last update: 2020-05-12 18:09:04

```

```

sonic# show bgp ipv6 unicast 5001:1111::/64 bestpath
BGP routing table entry for 5001:1111::/64
Paths: (1 available, best #1, table default)
 200
 1001:2222::2 from 1001:2222::2 (2.2.2.2)
 Origin IGP, valid, best
 Community: 600:755 noExport noAdvertise localAs noPeer
 Last update: 2020-05-12 18:09:04

```

### View IPv6 routes in BGP communities

**Syntax**

```
show bgp ipv6 unicast [vrf vrf-name] community {AA:NN | local-as | no-advertise | no-export | no-peer} [exact-match]
```

**Parameters**

- *vrf vrf-name* — (Optional) Enter a VRF instance name to filter the display of BGP IPv6 routes.
- *AA:NN* — Display IPv6 routes that match the specified community number, where *AA* is an autonomous system number from 1 to 65535; *NN* is a network number from 1 to 65535.
- *ipv6-prefix/mask* — Enter an IPv6 prefix or address and bit mask (1 to 32) to display routes only to matching BGP devices in the network.
- *local-as* — Display BGP routes in the local autonomous system.
- *no-advertise* — Display BGP routes in the no-advertise community — routes not advertised to any neighbor in the local or an external autonomous system.
- *no-export* — Display BGP routes in the no-export community — routes advertised only to neighbors in the same autonomous system.
- *no-peer* — Display BGP routes in the no-peer community — routes that may not be advertised to neighbors in the local or external autonomous systems.
- *exact-match* — (Optional) Display only the BGP routes that exactly match the specified community criteria.

**Example**

```
sonic# show bgp ipv6 unicast community 600:755
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Route status codes: * - valid, > - best
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
* 5001:1111::/64 fe80::92b1:1cff:fef4:ab9b 0 0 200 i
*> fe80::92b1:1cff:fef4:ab9b 0 0 200 i
* 5002:2222::/64 fe80::92b1:1cff:fef4:ab9b 0 0 200 i
*> fe80::92b1:1cff:fef4:ab9b 0 0 200 i
```

```
sonic# show bgp ipv6 unicast community 600:755 exact-match
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Route status codes: * - valid, > - best
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
* 5002:2222::/64 fe80::92b1:1cff:fef4:ab9b 0 0 200 i
*> fe80::92b1:1cff:fef4:ab9b 0 0 200 i
```

**Filter IPv6 routes in BGP routing table with route map****Syntax**

```
show bgp ipv6 unicast [vrf vrf-name] route-map map-name
```

**Parameters**

- *vrf vrf-name* — (Optional) Enter a VRF instance name to filter the display of BGP IPv6 routes.
- *route-map map-name* — Enter a route map to display only matching BGP routes.

**Example**

```
sonic# show bgp ipv6 unicast route-map map1
BGP routing table information for VRF default
Router identifier 120.0.0.1, local AS number 65001
Route status codes: * - valid, > - best
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight
Path
*> 2001:1:1::/64 fe80::1e72:1dff:fec4:2760 0 0 65100 ?
*> fd00::/80 fe80::1e72:1dff:fec4:2760 0 0 65100 ?
```

**Display BGP IPv6 route statistics****Syntax**

```
show bgp ipv6 unicast [vrf vrf-name] statistics
```

**Parameters**

- *vrf vrf-name* — (Optional) Enter a VRF instance name to filter the display of BGP IPv6 routes.

**Example**

```
sonic# show bgp ipv6 unicast statistics
BGP IPv4 Unicast RIB statistics
```

```

Total Advertisements : 5
Total Prefixes : 3
Average prefix length : 64.00
Unaggregateable prefixes : 3
Maximum aggregateable prefixes: 0
BGP Aggregate advertisements : 0
Address space advertised : 5.53402e+19
 % announced :
 /8 equivalent :
 /24 equivalent :
Advertisements with paths : 5
Longest AS-Path (hops) : 1
Average AS-Path length (hops) : 0.80
Largest AS-Path (bytes) : 6
Average AS-Path size (bytes) : 4.80
Highest public ASN : 200

```

## View BGP IPv6 routes in VRFs

**Syntax**      `show bgp ipv6 unicast {vrf-name | default | all}`

- Parameters**
- `vrf vrf-name` — (Optional) Enter a VRF instance name to filter the display of BGP IPv6 routes.
  - `default` — Display the BGP IPv6 routes in the default VRF.
  - `all` — Display the BGP IPv6 routes in all VRFs.

**Example**

```

sonic# show bgp ipv6 unicast vrf all
BGP routing table information for VRF default
Router identifier 1.1.1.1, local AS number 100
Route status codes: * - valid, > - best
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
*> 2211:2211::/64 :: 0 32768 i
* 5001:1111::/64 fe80::92b1:1cff:fef4:ab9b 0 200 i
*> fe80::92b1:1cff:fef4:ab9b 0 200 i
* 5002:2222::/64 fe80::92b1:1cff:fef4:ab9b 0 200 i
*> fe80::92b1:1cff:fef4:ab9b 0 200 i

BGP routing table information for VRF Vrf_blue
Router identifier 3.3.3.3, local AS number 300
Route status codes: * - valid, > - best
Origin codes: i - IGP, e - EGP, ? - incomplete
 Network Next Hop Metric LocPref Weight Path
*> 2112:3322::/64 :: 0 32768 i

```

## IPv4 Address Resolution Protocol

The Address Resolution Protocol (ARP) allows IPv4 packets to be sent across networks by translating IP network addresses to MAC hardware addresses, and MAC addresses to IP addresses.

The MAC address and corresponding IP address of destination devices are maintained in the ARP table. Using the IP address, the ARP table allows the switch to quickly retrieve the associated MAC address, encapsulate the IP packet in a L2 frame, and transmit it over the network to a destination.

 **NOTE:** For IPv6 address translation, Enterprise SONiC uses the Network Discovery Protocol (NDP).

## View IPv4 ARP entries

To display ARP table entries, use the `show ip arp` command. To filter the output, specify an interface, port channel, or VLAN, an IP address in the format `x.x.x.x`, a MAC address, or a combination of more than one value to match. You can also display summary information and the IPv4 ARP entries in a specified VRF.

```
show ip arp [interface {Eth slot/port[/breakout-port] [summary] | PortChannel number [summary] | Vlan vlan-id [summary] | Management port-number [summary]}] [ip-address [mac-address mac-address] [summary] [vrf vrf-name]]
```

```
sonic# show ip arp
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

192.168.1.4 00:01:02:03:44:55 Eth1/3 - Dynamic Fwd
192.168.2.4 00:01:02:03:ab:cd PortChannel1200 - Dynamic Fwd
192.168.3.6 00:01:02:03:04:05 Vlan100 Eth1/2 Dynamic Fwd
10.11.48.254 00:01:e8:8b:44:71 Management0 - Dynamic Fwd
10.14.8.102 00:01:e8:8b:44:71 Management0 - Dynamic Fwd
0.0.0.0 00:00:00:00:00:00 lo - Dynamic Fwd
```

**i** **NOTE:** The last ARP entry for the loopback IP address 0.0.0.0 may be entered from the SONiC NEIGH\_TABLE of APPL\_DB. To delete the entry, you must manually delete it from the REDIS\_DB using the redis-cli or any other application that allows you to edit the REDIS\_DB.

```
sonic# show ip arp interface Vlan 20
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

20.0.0.2 90:b1:1c:f4:9d:ba Vlan20 Eth1/1 Dynamic Fwd
20.0.0.5 00:11:22:33:44:55 Vlan20 Eth1/1 Dynamic Fwd
```

```
sonic# show ip arp 20.0.0.2
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

20.0.0.2 90:b1:1c:f4:9d:ba Vlan20 Eth1/1 Dynamic Fwd
```

```
sonic# show ip arp mac-address 90:b1:1c:f4:9d:ba
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

20.0.0.2 90:b1:1c:f4:9d:ba Vlan20 Eth1/1 Dynamic Fwd
```

```
sonic# show ip arp summary
Total Entries

2
```

```
sonic# show ip arp vrf Vrf_1
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

20.0.0.2 90:b1:1c:f4:9d:ba Vlan20 Eth1/1 Dynamic Fwd
20.0.0.5 00:11:22:33:44:55 Vlan20 Eth1/1 Dynamic Fwd
```

## Clear IPv4 entries

To delete dynamically learned IPv4 entries from the ARP table, use the `clear ip arp` command. To specify the entries to be deleted, enter an interface, port channel, or VLAN, an IPv4 address, or a combination to match. Enter `force` to delete statically configured ARP entries. Use the `show ip arp` command to verify that the IPv4 entries have been deleted.

```
clear ip arp [interface {Eth slot/port[/breakout-port] | PortChannel number | Vlan vlan-id | Management port-number}] [ip-address] [vrf vrf-name]
```

```
sonic# show ip arp
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action
192.168.1.4 00:01:02:03:44:55 Eth1/3 - Dynamic Fwd
192.168.2.4 00:01:02:03:ab:cd PortChannel1200 - Dynamic Fwd
192.168.3.6 00:01:02:03:04:05 Vlan100 Eth1/2 Dynamic Fwd
10.11.48.254 00:01:e8:8b:44:71 Management0 - Dynamic Fwd
10.14.8.102 00:01:e8:8b:44:71 Management0 - Dynamic Fwd

sonic# clear ip arp interface Vlan 100
sonic# show ip arp
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action
192.168.1.4 00:01:02:03:44:55 Eth1/3 - Dynamic Fwd
192.168.2.4 00:01:02:03:ab:cd PortChannel1200 - Dynamic Fwd
10.11.48.254 00:01:e8:8b:44:71 Management0 - Dynamic Fwd
10.14.8.102 00:01:e8:8b:44:71 Management0 - Dynamic Fwd
```

### Protect CPU from unresolved IPv4 ARP entries

When a switch receives unresolved IP packets, the packets are trapped by the CPU for further processing. The switch proceeds to process each packet by trying to resolve its destination MAC address.

If a high number of packets that are destined for an unresolved IP are received in a small burst, each packet is trapped by the CPU causing CPU overload. To protect the CPU from such IP attacks, use ARP protection. ARP protection creates blackhole routes for unresolved IP destinations.

When a blackhole route is installed, traffic destined to the unresolved IP is dropped and not trapped by the CPU, creating no load on the CPU. In addition, the switch periodically tries to resolve the unreachable IP destinations.

To enable ARP protection for unresolved subnet traffic, configure the blackhole aging interval (60-14400 seconds; default 300) for neighbor traffic.

```
ip drop-neighbor aging-time seconds
```

For example:

```
sonic# ip drop-neighbor aging-time 12000
```

The `show ip arp` output indicates the routes that are dropped to stop traffic for unresolved ARP entries from affecting the CPU. The traffic on dropped routes is forwarded as soon as ARP resolves the addresses. To save ARP table space, a dropped blackholed entry is removed from hardware after the blackhole ageing interval expires.

```
sonic# show ip arp

Address Hardware address Interface Egress Interface Action
11.0.0.2 80:a2:35:26:45:5e Ethernet68 - Fwd
10.0.0.2 00:00:00:00:00:00 Ethernet64 - Drop
10.59.128.2 18:80:90:23:98:49 Management0 - Fwd
```

## View IPv6 NDP entries

To view MAC addresses that correspond to IPv6 addresses in the neighbor discovery protocol (NDP) table, use the `show ipv6 neighbors` command. To filter the output, specify an interface, port channel, or VLAN, or an IPv6 address in the format `A::B`, or a combination to match. You can also view summary information and the IPv6 NDP entries in a specified VRF.

```
sonic# show ipv6 neighbors [interface {Eth slot/port[/breakout-port] [summary] | PortChannel number [summary] | Vlan vlan-id [summary]}] [ipv6-address] [mac-address mac-address] [summary] [vrf vrf-name]
```

```
sonic# show ipv6 neighbors
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)
```

| Address                   | Hardware address  | Interface   | Egress Interface | Type    | Action |
|---------------------------|-------------------|-------------|------------------|---------|--------|
| fe80::6f8:f8ff:fe6b:a91   | 04:f8:f8:6b:0a:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:c91   | 04:f8:f8:6b:0c:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:d91   | 04:f8:f8:6b:0d:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:e91   | 04:f8:f8:6b:0e:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:1691  | 04:f8:f8:6b:16:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:3891  | 04:f8:f8:6b:38:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::1a5a:58ff:fe66:8020 | 18:5a:58:66:80:20 | Management0 | -                | Dynamic | Fwd    |
| fe80::1a5a:58ff:fe66:a320 | 18:5a:58:66:a3:20 | Management0 | -                | Dynamic | Fwd    |
| fe80::1e72:1dff:feel:1e7f | 1c:72:1d:e1:1e:7f | Management0 | -                | Dynamic | Fwd    |
| fe80::1e72:1dff:feel:247f | 1c:72:1d:e1:24:7f | Management0 | -                | Dynamic | Fwd    |
| fe80::1e72:1dff:feel:777f | 1c:72:1d:e1:77:7f | Management0 | -                | Dynamic | Fwd    |
| fe80::1e72:1dff:fee:1dff  | 1c:72:1d:ee:1d:ff | Management0 | -                | Dynamic | Fwd    |
| fe80::5054:ff:fe69:392b   | 52:54:00:69:39:2b | Management0 | -                | Dynamic | Fwd    |
| fe80::82a2:35ff:fe26:4e5e | 80:a2:35:26:4e:5e | Eth1/3      | -                | Dynamic | Fwd    |
| fe80::82a2:35ff:fef2:79b8 | 80:a2:35:f2:79:b8 | Eth1/4      | -                | Dynamic | Fwd    |
| fe80::82a2:35ff:fef2:7d3c | 80:a2:35:f2:7d:3c | Eth2/1      | -                | Dynamic | Fwd    |
| fe80::82a2:35ff:fef2:8a20 | 80:a2:35:f2:8a:20 | Eth2/2      | -                | Dynamic | Fwd    |
| fe80::82a2:35ff:fef2:8c78 | 80:a2:35:f2:8c:78 | Eth2/3      | -                | Dynamic | Fwd    |
| fe80::923c:b3ff:fec5:95ba | 90:3c:b3:c5:95:ba | Management0 | -                | Dynamic | Fwd    |
| fe80::923c:b3ff:fec5:bd92 | 90:3c:b3:c5:bd:92 | Management0 | -                | Dynamic | Fwd    |
| fe80::ba6a:97ff:fee2:5e9c | b8:6a:97:e2:5e:9c | Management0 | -                | Dynamic | Fwd    |
| fe80::e201:a6ff:fedf:12b0 | e0:01:a6:df:12:b0 | Management0 | -                | Dynamic | Fwd    |
| fe80::e201:a6ff:fef3:8080 | 00:00:00:00:00:00 | Management0 | -                | Dynamic | Drop   |

```
sonic# show ipv6 neighbors fe80::6f8:f8ff:fe6b:e91
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)
```

| Address                 | Hardware address  | Interface | Egress Interface | Type    | Action |
|-------------------------|-------------------|-----------|------------------|---------|--------|
| fe80::6f8:f8ff:fe6b:e91 | aa:bb:cc:dd:ee:ff | Eth1/3    | -                | Dynamic | Fwd    |

```
sonic# show ipv6 neighbors mac-address 52:54:00:69:39:2b
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)
```

| Address                  | Hardware address  | Interface   | Egress Interface | Type    | Action |
|--------------------------|-------------------|-------------|------------------|---------|--------|
| fe80::e6f0:4ff:fe79:34c7 | 52:54:00:69:39:2b | Management0 | -                | Dynamic | Fwd    |

```
sonic# show ipv6 neighbors interface Management 0
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)
```

| Address                   | Hardware address  | Interface   | Egress Interface | Type    | Action |
|---------------------------|-------------------|-------------|------------------|---------|--------|
| fe80::6f8:f8ff:fe6b:a91   | 04:f8:f8:6b:0a:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:c91   | 04:f8:f8:6b:0c:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:d91   | 04:f8:f8:6b:0d:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:e91   | 04:f8:f8:6b:0e:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:1691  | 04:f8:f8:6b:16:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::6f8:f8ff:fe6b:3891  | 04:f8:f8:6b:38:91 | Management0 | -                | Dynamic | Fwd    |
| fe80::1a5a:58ff:fe66:8020 | 18:5a:58:66:80:20 | Management0 | -                | Dynamic | Fwd    |
| fe80::1a5a:58ff:fe66:a320 | 18:5a:58:66:a3:20 | Management0 | -                | Dynamic | Fwd    |

```
fe80::1e72:1dff:fe1:1e7f 1c:72:1d:e1:1e:7f Management0 - Dynamic Fwd
...
```

### Clear IPv6 entries

To delete dynamically learned IPv6 entries from the NDP table, use `clear ipv6 neighbors`. To specify the entries to delete, enter an interface, port channel, or VLAN, an IPv6 address, or a combination to match. Enter `force` to delete statically configured ARP entries. Use `show ipv6 neighbors` to verify that the IPv6 entries have been deleted.

```
sonic# clear ipv6 neighbors [interface {Eth slot/port[/breakout-port] | PortChannel number | Vlan vlan-id | Management port-number}] [ipv6-address] [vrf vrf-name]
```

```
sonic# show ipv6 neighbors
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

20::2 aa:bb:cc:dd:ee:ff Eth1/3 - Dynamic Fwd
fe80::e6f0:4ff:fe79:34c7 e4:f0:04:79:34:c7 Management0 - Dynamic Fwd

sonic# clear ipv6 neighbors 20::2
sonic# show ipv6 neighbors
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

fe80::e6f0:4ff:fe79:34c7 e4:f0:04:79:34:c7 Management0 - Dynamic Fwd

sonic# clear ipv6 neighbors
sonic# show ipv6 neighbors
```

### Protect CPU from unresolved IPv6 ARP entries

In the `show ipv6 neighbors` output, the Action column indicates if packets from an entry in the NDP table are to be treated normally and forwarded or if they are dropped.

```
sonic# show ipv6 neighbors
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

11::1 80:a2:35:26:45:5e Ethernet68 - Dynamic Fwd
11::3 00:00:00:00:00:00 Ethernet64 - Dynamic Drop
```

## Open Shortest Path First

Open Shortest Path First Protocol (OSPF) is a link state interior gateway routing protocol (IGRP) as described in RFC2328.

OSPF describes link-state information in a message known as a Link State Advertisement (LSA), which is then propagated through to all other routers in a link-state routing domain, by a process called flooding. Each OSPF router thus builds up a Link State Database (LSDB) of all the link-state messages. From this collection of LSAs in the LSDB, each router can then calculate the shortest path to any other router, based on some common metric, by using Edgar Dijkstras Shortest Path First algorithm.

Enterprise SONiC uses FRR packages for running routing protocols. OSPFv2 is also adapted from a customized FRR software package. The OPSFv2 routing daemon resides within the BGP docker container along with other routing protocol daemons, such as BGP, static route.

OSPFv2 capabilities supported:

- OSPF configuration on Ethernet, loopback, VLAN, and port-channel IPv4 interfaces.
- OSPFv2 configuration on default and user-defined VRFs.
- Multiple OSPF areas and stub areas.
- Type-1 to Type-5 LSAs .
- Virtual links and Passive interfaces.
- BFD on OSPF interface sessions.
- Plain text and message digest (MD) password encryption.

- Type-3 Summary LSA prefix filtering and substitution.
- Route redistribution into OSPFv2, from route type BGP, static, connected, kernel, and default-route.
- Route-map based filtering in route redistribution.
- OSPF ECMP routes.
- 50K external route and 5K internal route prefix.

## Enable OSPFv2

Enable OSPFv2 by configuring OSPF router within a VRF. Use this command to configure OSPFv2 within a VRF. Configuring OSPF changes the mode to OSPF router configuration mode.

- Enable OSPF.

```
[no] router ospf [vrf vrf-name]
```

*vrf-name* — VRF name string

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)#
sonic(config)# router ospf vrf Vrf-blue
sonic(conf-router-ospf)#
```

View the OSPF router details:

```
sonic # show ip ospf vrf default
VRF Name: default
OSPF Routing Process, Router ID: 10.1.1.1
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 0 millisec(s)
Minimum hold time between consecutive SPFs 50 millisec(s)
Maximum hold time between consecutive SPFs 5000 millisec(s)
Hold time multiplier is currently 1
SPF algorithm last executed 0h0m8s ago
Last SPF duration 56 usecs
SPF timer is inactive
LSA minimum interval 5000 msec
LSA minimum arrival 1000 msec
Write Multiplier set to 20
Refresh timer 10 secs
Number of external LSA 0. Checksum Sum 0x00000000
Number of opaque AS LSA 0. Checksum Sum 0x00000000
Number of areas attached to this router: 1
Area ID: 0.0.0.1
 Shortcutting mode: Default, S-bit consensus: no
 Number of interfaces in this area: Total: 4 , Active: 4
 Number of fully adjacent neighbors in this area: 4
 Area has simple password authentication
 Number of full virtual adjacencies going through this area: 0
 SPF algorithm executed 13 times
 Number of LSA 6
 Number of router LSA 2. Checksum Sum 0x0000f122
 Number of network LSA 4. Checksum Sum 0x00032724
 Number of summary LSA 0. Checksum Sum 0x00000000
 Number of ASBR summary LSA 0. Checksum Sum 0x00000000
 Number of NSSA LSA 0. Checksum Sum 0x00000000
 Number of opaque link LSA 0. Checksum Sum 0x00000000
 Number of opaque area LSA 0. Checksum Sum 0x00000000
```

## Enable OSPF on Interfaces

Before enabling OSPF on an interface, configure an Ipv4 interface with an IP address on it and bind the interface to the required VRF. In order to enable OSPF on an IPv4 interface, you must associate an OSPF Area-ID with the interface. You can enable OSPF on Ethernet, VLAN, Port Channel, and Loopback interfaces.

Enable OSPF sessions between two OSPF routers by enabling OSPF on the interfaces connecting them. Such connecting interface shall reside within the VRF where the OSPF router is configured and shall belong to the same Area-ID.

Enable OSPF on an interface using these two configuration types:

- By configuring OSPF area under interface configuration mode:

```
ip ospf area area-id
```

- By binding or associating the network address of an interface to an OSPF area under OSPFv2 router configuration mode:

```
network network-prefix area area-id
```

Within a VRF, user shall either use one of the above config types, that is, both config type configurations is not allowed at a time within a VRF.

Use this interface mode command to enable or disable OSPF on an IPv4 interface. Area ID can be any 32-bit unsigned integer number, in decimal format, or dotted Ipv4 like format.

### Example

```
sonic(config)# interface Eth1/15
sonic(conf-if-Eth1/15)# ip address 10.10.3.1/24
sonic(conf-if-Eth1/15)# ip ospf area 0

sonic(config)# interface Eth1/16
sonic(conf-if-Eth1/16)# ip vrf forwarding Vrf-blue
sonic(conf-if-Eth1/16)# ip ospf area 1 10.1.1.1

sonic(config)# interface Eth1/17
sonic(conf-if-Eth1/17)# ip vrf forwarding Vrf-red
sonic(conf-if-Eth1/17)# ip ospf area 2 10.2.2.1

sonic(config)# interface vlan 10
sonic(conf-if-Eth1/17)# ip ospf area 10

sonic(config)# interface Portchannel 16
sonic(conf-if-Eth1/17)# ip vrf forwarding Vrf-blue
sonic(conf-if-Eth1/17)# ip ospf area 0
```

Use this router mode command to enable or disable OSPF on an IPv4 interface:

```
sonic(config)# router ospf
sonic(conf-router-ospf)# network 10.10.3.0/24 area 0
sonic(conf-router-ospf)# network 10.3.3.0/24 area 0
sonic(conf-router-ospf)# network 192.3.3.0/24 area 2
sonic(conf-router-ospf)# network 168.2.1.0/24 area 1

sonic(conf)# router ospf vrf Vrf-blue
sonic(conf-router-ospf)# network 10.1.1.0/24 area 0.0.0.1

sonic(config)# router ospf vrf Vrf-red
sonic(conf-router-ospf)# network 10.2.2.0/24 area 0.0.0.2
```

When using network command to associate an interface to an OSPF area, IPv4 address Prefix length in interface command must be equal or bigger (that is, smaller network) than prefix length in network statement.

View the OSPF sessions:

```
show ip ospf vrf vrf-name neighbor [detail]
```

## Example

| Neighbor ID | Pri | State        | Dead Time | Address   | Interface              | RXmtL | RqstL | DBsmL |
|-------------|-----|--------------|-----------|-----------|------------------------|-------|-------|-------|
| 10.1.1.2    | 1   | Full/DROther | 31.246s   | 10.10.3.2 | Eth1/15:10.10.3.1      | 0     | 0     | 0     |
| 10.1.1.2    | 1   | Full/DROther | 36.028s   | 10.10.6.2 | PortChannel1:10.10.6.1 | 0     | 0     | 0     |
| 10.1.1.2    | 1   | Full/DROther | 32.801s   | 10.10.4.2 | Vlan2:10.10.4.1        | 0     | 0     | 0     |
| 10.1.1.2    | 1   | Full/DROther | 34.423s   | 10.10.5.2 | Vlan3:10.10.5.1        | 0     | 0     | 0     |

## Configure OSPF router ID

Configure OSPFv2 Router Identifier explicitly for every OSPF router within a VRF. Router ID configuration is optional. If you configure a router ID, the software chooses that router ID as the OSPF router ID.

Whenever router ID is not configured, router ID selection happens as per below preference.

- Most recently used router id value; (this can happen when user unconfigures router ID).
- FRR recommended value of Router Id. FRR chooses router id in below order.
  - FRR global mode configured router id value, if any.
  - Highest IPv4 address value among SONiC physical and Loopback interface IPv4 addresses.

Use this router mode command to configure or unconfigure OSPF router ID. Router ID can be any 32-bit unsigned integer number, in decimal format, or dottedIpv4 like format. OSPF router ID must be unique within the entire OSPF domain.

```
[no] ospf router-id router-id
```

*router-id*: OSPF router ID in decimal or dotted format.

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# ospf router-id 10.1.1.1
```

## Configure OSPF area level authentication type

You can configure OSPFv2 authentication type per area. When authentication is configured, all interfaces that are configured within that area use the configured authentication type. If there is any interface mode authentication type configured then that interface mode authentication type takes precedence over area level config.

Authentication type is none, when it is not configured by user. User can enable plain text authentication type or Message Digest type authentication. Whenever authentication type is configured for an area, user shall configure corresponding authentication keys (passwords) at all the OSPF interfaces belonging to that area.

Use this router mode command to configure or unconfigure OSPF authentication for an area.

```
[no] area area-id authentication [message-digest]
```

*area-id* — OSPF area ID in decimal or dotted format

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# ospf 0.0.0.0 authentication
sonic(conf-router-ospf)# ospf 0.0.0.1 authentication message-digest
```

## Configure OSPF interface level authentication type and keys

Configure the OSPFv2 authentication type and authentication keys or passwords for individual OSPF interfaces.

Authentication type is none, when it is not configured by user. User can enable plain text authentication type or Message Digest type authentication. Whenever authentication type is configured for an interface, user shall configure corresponding authentication keys (passwords) for that OSPF interface.

Plain text authentication can be up to eight characters long. Message Digest (MD5) authentication key can be up to 16 character long. MD5 authentication type can accept up to 255 authentication keys per interface and interface IP. Every MD5 authentication key is uniquely identified by an authentication key-id with value range in 1 and 255. Authentication Keys are saved in an encrypted form.

Use this interface mode command to configure or unconfigure OSPF message authentications.

```
[no] ip ospf authentication [null | message-digest] [if-ip-addr]
[no] ip ospf authentication-key key [if-ip-addr]
[no] ip ospf message-digest-key key-id md5 key [if-ip-addr]
```

- *If-ip-addr* — Interface IPv4 address
- *key* — Authentication key password (up to 8 or 16 characters)
- *key-id* — MD5 authentication key Identifier (1 to 255)
- *if-ip-addr* — Interface IP address

#### Example

```
sonic(config)# interface Eth1/15
sonic(conf-if-Eth1/15)# ip ospf authentication
sonic(conf-if-Eth1/15)# ip ospf authentication-key ospfpswd

sonic(config)# interface Eth1/17
sonic(conf-if-Eth1/17)# ip ospf authentication 10.10.3.2
sonic(conf-if-Eth1/17)# ip ospf authentication-key ospfpswd 10.10.3.2

sonic(config)# interface Eth1/16
sonic(conf-if-Eth1/16)# ip ospf authentication message-digest
sonic(conf-if-Eth1/16)# ip ospf message-digest-key 1 md5 ospfpswd1
sonic(conf-if-Eth1/16)# ip ospf message-digest-key 2 md5 ospfpswd2
sonic(conf-if-Eth1/16)# ip ospf message-digest-key 9 md5 ospfpswd9

sonic(config)# interface Eth1/18
sonic(conf-if-Eth1/18)# ip ospf authentication null
```

## Configure OSPF passive interfaces

Configure OSPFv2 passive interfaces using router mode configurations. On a passive interface, OSPF does not trigger OSPF hello or initiate OSPF sessions. Passive interfaces are advertised as a stub link in the router-LSA.

To configure all OSPF interfaces as passive interfaces by default, enter the `passive-interface default` option. To reactivate all or specified interfaces, use the `no passive-interface` command.

```
sonic(conf-router-ospf)# [no] passive-interface {default | Ethslot/port [ip-addr]}
```

- *Ethslot/port* — Interface name
- *ip-addr* — Interface IP address

#### Example

```
sonic(config)# router ospf vrf Vrf-blue
sonic(conf-router-ospf)# passive-interface Eth1/15
sonic(conf-router-ospf)# passive-interface Eth1/17 10.3.10.1
```

```
sonic(config)# router ospf vrf Vrf-red
sonic(conf-router-ospf)# passive-interface default
sonic(conf-router-ospf)# no passive-interface Eth1/16
```

## Configure OSPF virtual links

OSPFV2 Virtual links are used to connect backbone routers across a nonbackbone area. The area through which the virtual link is configured, known as transit area, must have full routing information. The transit area cannot be a stub area. You must configure virtual links on both end backbone routers.

Configure virtual links using OSPF router mode configurations as below. Single virtual link command provides options to configure all parameters that are related to the Virtual link.

Virtual links can have clear text password, message-digest based passwords or no password configured at all. When clear text and message digest password is configured, corresponding authentication-key or message-digest-key parameters must be configured.

Authentication key (password) is saved in encrypted form in configurations. User shall always provide actual password while configuring authentication keys.

```
[no] area area-id virtual-link remote-id [authentication [null | message-digest] | authentication-key key | message-digest-key key-id md5 key | dead-interval time-value | hello-interval time-value | retransmit-interval time-value | transmit-delay time-value]
```

- *area-id* — OSPF area ID in decimal or dotted format
- *remote-id* — Remote router ID in dotted format
- *key* — Authentication key password (up to 8 or 16 characters)
- *key-id* — MD5 authentication key identifier (1 to 255)
- *time-value* — Time interval value in seconds (1 to 65535)

#### Example

```
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 authentication
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 authentication-key mypasswd
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 authentication message-digest
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 message-digest-key 19 md5
md5password19
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 message-digest-key 20 md5
md5password20
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 authentication null
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 dead-interval 60
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 hello-interval 20
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 retransmit-interval 15
sonic(conf-router-ospf)# area 19 virtual-link 1.1.1.9 transmit-delay 10
```

Prefix length in interface must be equal or bigger (that is, smaller network) than prefix length in network statement.

## Configure OSPF ABR type

OSPF router ABR can be of type Cisco, IBM, shortcut or standard. The "Cisco" and "IBM" types are equivalent. The OSPF standard for ABR behavior does not allow an ABR to consider routes through nonbackbone areas when its links to the backbone are down, even when there are other ABRs in attached nonbackbone areas which still can reach the backbone - this restriction exists primarily to ensure routing-loops are avoided.

With the "Cisco" or "IBM" ABR type, the default in this release, this restriction is lifted, allowing an ABR to consider summaries learned from other ABRs through nonbackbone areas, and hence route through nonbackbone areas as a last resort when, and only when, backbone links are down.

The `ospf abr-type` command is used to configure or unconfigure ABR type as below.

```
[no] ospf abr-type [isco | ibm | shortcut | standard]
```

#### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# ospf abr-type standard
```

## Configure OSPF area shortcuts

OSPF Backbone area is responsible for routing distribution between nonbackbone areas. Backbone area should be contiguous, but it does not always imply a physical adjacency. You can achieve backbone area router connections using virtual connections.

By configuring the shortcut type, you can enable or disable shortcut routes to backbone area. When the shortcut type is default then the area is used for shortcircuiting only if ABR does not have a link to the backbone area or this link was lost. When shortcut type is 'enable' then the area is used for shortcircuiting every time the route that goes through it is cheaper. When shortcut type is 'disable' then the area is never used by ABR for routes shortcircuiting.

Configure or unconfigure OSPFv2 area shortcut types under OSPF router configuration mode.

```
[no] area area-id shortcut {default | enable | disable}
```

- *area-id* — OSPF area ID in decimal or dotted format

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# area 1 shortcut enable
```

## Configure OSPF RFC compatibility

OSPFv2 RFC2328, the successor to RFC1583, suggests according to section G.2 (changes) in section 16.4 a change to the path preference algorithm that prevents possible routing loops that were possible in the old version of OSPFv2. More specifically it demands that interarea paths and intra-area backbone path are now of equal preference but still both preferred to external paths.

Enable the OSPF RFC1583 compatibility:

```
[no] ospf rfc1583compatibility
```

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# ospf rfc1583compatibility
```

## Configure OSPF adjacency logging

Enable OSPFv2 adjacency state logs by configuring adjacency logs. With the optional detail argument, all changes in adjacency status are shown.

Enable OSPF adjacency logs:

```
[no] log-adjacency-changes [detail]
```

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# log-adjacency-changes
```

## Configure OSPF LSA timers

Configure OSPFv2 LSA refresh interval, minimum interval, and throttle timer:

```
[no] refresh timer refresh-time
[no] timers lsa min-arrival min-arr-time
[no] timers throttle lsa all throttle-time
```

- *refresh-time* — Refresh time (10 to 1800 seconds)
- *min-arr-time* — Minimum arrival time (0 to 600000 milliseconds)

- *throttle-time* — Throttle time (0 to 5000 milliseconds)

#### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# refresh timer 40
sonic(conf-router-ospf)# timers lsa min-arrival 30
sonic(conf-router-ospf)# timers throttle lsa all 150
```

## Configure OSPF SPF throttle timers

OSPFv2 SPF algorithm throttle timers set initial-delay, the initial-hold-time and the maximum-hold-time between when SPF is calculated and the event which triggered the calculation. The times are specified in milliseconds and must be in the range of 0 to 600000 milliseconds.

The initial-delay specifies the minimum amount of time to delay SPF calculation. Hence it affects how long SPF calculation is delayed after an event which occurs outside of the hold-time of any previous SPF calculation, and also serves as a minimum hold-time.

Consecutive SPF calculations is always separated by at least 'hold-time' milliseconds. The hold-time is adaptive and initially is set to the initial-hold-time configured with the throttle command. Events which occur within the hold-time of the previous SPF calculation causes the hold-time to be increased by initial-hold-time, bounded by the maximum-hold-time configured with throttle command. If the adaptive hold-time elapses without any SPF-triggering event occurring then the current hold-time is reset to the initial-hold-time. You can view the current hold-time using the `show ip ospf` command, where it is expressed as a multiplier of the initial-hold-time.

Configure OSPF SPF throttle timer values using this OSPF router mode command:

```
[no] timers throttle spf initial-delay initial-hold-time max-hold-time
```

- *initial-delay* — Time value (0 to 600000 milliseconds)
- *initial-hold-time* — Time value (0 to 600000 milliseconds)
- *max-hold-time* — Time value (0 to 600000 milliseconds)

#### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# timers throttle spf 200 400 10000
```

In the above example, the initial-delay is set to 200ms, the initial-hold-time is set to 400ms and the maximum-hold-time to 10s. Hence there is always at least 200ms between an event (which requires SPF calculation) and the SPF calculation. Further consecutive SPF calculations are always separated by between 400ms to 10s, the hold-time increases by 400ms each time an SPF-triggering-event occurs within the hold-time of the previous SPF calculation.

## Configure OSPF max-metric advertising

OSPFv2, per RFC 313, describes its transit links in its router-LSA as having infinite distance (max-metric) so that other routers avoid calculating transit paths through the router while still being able to reach networks through the router.

This support may be enabled administratively (that is, indefinitely) or conditionally. Conditional enabling of max-metric router LSAs can be for a period of seconds after start-up.

Enabling this for a period after start-up allows OSPF to converge fully first without affecting any existing routes used by other routers, while still allowing any connected stub links and/or redistributed routes to be reachable.

Enabling this feature administratively allows for administrative intervention for whatever reason, for an indefinite period of time. Note that if the configuration is saved then this administrative form of the stub-router command is also saved. If system or docker is restarted later, the command then takes effect until it is manually unconfigured.

Configure or unconfigure the OSPF max metric feature:

```
[no] max-metric router-lsa administrative
[no] max-metric router-lsa on-startup time-value
```

- *time-value* — Time value (5 to 86400 seconds)

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# max-metric router-lsa on-startup 60
```

## Configure OSPF route distances

Assign OSPFv2 calculated routes with user configured routing distances within a router. You can configure the distance value on all OSPFv2 generated routes. Distance value configurations can also be done based on the source of OSPF route, like intra-area route, interarea route and external route with respect to current router.

Configure OSPFv2 route distance under the OSPF router configuration mode:

```
[no] distance distance-value
[no] distance ospf intra-area distance-value
[no] distance ospf inter-area distance-value
[no] distance ospf external distance-value
```

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# distance 25
sonic(conf-router-ospf)# distance ospf inter-area 30
sonic(conf-router-ospf)# distance ospf external 60
```

## Configure OSPF auto cost reference bandwidth

OSPFv2 calculates route costs based on OSPF interface costs. Interface costs can either be manually configured or calculated automatically. Manually configured interface cost takes precedence over auto calculated interface cost.

Interface cost auto calculation is by considering a reference bandwidth and interface/link bandwidth. Cost of reference bandwidth is considered to be having cost 1. Default reference bandwidth is 100 Gigabytes. Link bandwidth is chosen in the preference order of configured link bandwidth or link actual speed or default link bandwidth (10G). Link cost is calculated as below.

Link cost = (Reference bandwidth) / (Link bandwidth + 0.5)

Calculated link cost can be of less than 1, in such a case it is always rounded to link cost 1.

Configure or unconfigure the OSPFv2 auto cost reference bandwidth under OSPF router configuration mode:

```
[no] auto-cost reference-bandwidth ref-bandwidth
```

- *ref-bandwidth* — Reference bandwidth (1 to 4294967 megabits)

### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# auto-cost reference-bandwidth 10000
```

## Configure OSPF stub area and its parameters

Configure the OSPFv2 area as a stub area with or without summary routes. Stub area is an area where no router originates routes external to OSPF. Thus this is an area where all external routes are considered reachable through ABRs. Hence, ABRs for such an area do not need to pass AS-External LSAs or ASBR-Summary LSAs into the area. They need to pass only the Network-Summary LSAs into such an area, along with a default-route summary. No summary stub area prevents the ABR from even injecting interarea summaries into the specified stub area.

Configure or unconfigure the OSPFv2 Stub area under OSPF router configuration mode:

```
[no] area area-id stub [no-summary]
```

- *area-id* — OSPF area ID in decimal or dotted format

## Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# area 0.0.0.1 stub
sonic(conf-router-ospf)# area 0.0.0.3 stub no summary
```

Configure the OSPFv2 stub area summary default cost under OSPF router configuration mode:

```
[no] area area-id default-cost cost-value
```

- *Cost-value* — Default cost (0 to 1677721)

## Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# area 0.0.0.1 default-cost 30
```

## Configure OSPF inter area summary route filters

Manage the OSPFv2 inter area Summary LSA route propagation at an ABR using prefix lists and range lists.

Use the *in* prefix list to filter out incoming routes into an area at an ABR. Similarly use the *out* prefix list to filter out outgoing routes from an area.

Range lists can also be used to manage advertising of prefixes from an area. Using range lists it is possible to modify the prefix as well as cost or the route prefix to be advertised. This option summarizes intra area paths from specified area into one Type-3 summary-LSA announced to other areas. You can use this configuration only in ABR and you can summarize only router-LSAs and network-LSAs.

Configure or unconfigure the OSPFv2 inter area route propagation prefix filtering under OSPF router configuration mode:

```
[no] area area-id filter-list prefix prefix-list in
[no] area area-id filter-list prefix prefix-list out
```

- *area-id* — OSPF area ID in decimal or dotted format
- *prefix-list* — IPv4 prefix list name

## Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# area 0 filter-list prefix a0inplst in
sonic(conf-router-ospf)# area 0 filter-list prefix a0output out
```

Configure or unconfigure the OSPFv2 inter area route propagation range list under OSPF router configuration mode:

```
[no] area area-id range ip-prefix
[no] area area-id range ip-prefix not-advertise
[no] area area-id range ip-prefix cost cost-value
[no] area area-id range ip-prefix advertise cost cost-value
[no] area area-id range ip-prefix substitute sub-ip-prefix
```

## Configure OSPF route redistribution

OSPFv2 can redistribute external routes into OSPF routing domain. Redistribute BGP routes, Static routes, connected routes, and kernel routes into OSPF routing domain. Manage route redistribution using route maps in addition to explicitly specified cost and metric type.

Configure or unconfigure the OSPFv2 external route redistribution under OSPF router configuration mode:

```
[no] redistribute {kernel | connected | static| bgp} [metric metric-value| metric-type metric-type-value | route-map rmap-name]
```

- *metric-value* — Route cost to applied on route (0 to 16777214)
- *metric-value* — Route cost to applied on route (0 to 16777214)

- *metric-type-value* — Metric type (1 and 2)
- *rmap-name* — Name of the route map that have to be applied on routes

#### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# redistribute bgp
sonic(conf-router-ospf)# redistribute static metric 20 metric-type 1
sonic(conf-router-ospf)# redistribute bgp route-map ospf-rmap
sonic(conf-router-ospf)# redistribute bgp metric 20 metric-type 1 route-map ospfrmap
```

Configure the OSPFv2 default redistribution cost or metric under OSPF router configuration mode:

```
[no] default-metric metric-value
```

- *metric-value* — Route cost to be applied on route (0 to 16777214)

#### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# default-metric 30
```

## Configure OSPF default route origination

Default routes can be originated into OSPF routing domain. This option originates an AS-External LSA describing a default route into all external-routing capable areas, of the specified metric and metric type. If the 'always' option is specified then the default route is always advertised, even when there is no default route present in the routing table.

Configure the OSPFv2 default route origination under OSPF router configuration mode:

```
[no] default-information originate [{ always | metric metric-value | metric-type metric-type-value | route-map rmap-name }]
```

- *metric-value* — Route cost to applied on route (0 to 16777214)
- *metric-type-value* — Metric type (1 and 2)
- *rmap-name* — Name of the route map that have to be applied on routes

#### Example

```
sonic(config)# router ospf
sonic(conf-router-ospf)# default-information originate
sonic(conf-router-ospf)# default-information originate always
sonic(conf-router-ospf)# default-information originate route-map ospf-rmap
sonic(conf-router-ospf)# default-information originate metric 20 metric-type 1 route-map ospfrmap
```

## Configure OSPF interface parameters

Configure OSPF interface parameters under SONiC interface configuration mode. Interface configuration includes:

- Area association to an interface
- Interface type, MTU
- Message Authentication parameters
- Session timer interval parameters
- BFD

Interface parameters can also be associated with a specific interface address of the interface by specifying the interface IPv4 address. When interface address is specified, such a configuration parameter is applicable to only the OSPF session associated with the corresponding interface address.

Configure the OSPFv2 interface area association under OSPF interface configuration mode:

```
[no] ip ospf area area-id if-ip-addr
```

- *area-id* — OSPF area ID in decimal or dotted format
- *if-ip-addr* — Interface IP address

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf area 10
sonic(conf-if-Eth1/1)#
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# ip ospf area 0.0.0.0 10.3.1.2
```

Configure the OSPFv2 interface network type under OSPF interface configuration mode. OSPF interface network is of network type broadcast by default.

```
[no] ip ospf network {broadcast | point-to-point}
```

#### Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf network point-to-point
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# ip ospf network broadcast
```

Configure the OSPFv2 interface session priority parameters under OSPF interface configuration mode. The router with the highest priority is more eligible to become Designated Router. Setting the value to 0, makes the router ineligible to become Designated Router. The default value is 1.

```
[no] ip ospf priority priority-value [if-ip-addr]
```

- *priority-value* — Session priority (0 to 255)
- *if-ip-addr* — Interface IP address

#### Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf priority 10
```

Configure OSPFv2 interface session authentication parameters under OSPF interface configuration mode:

```
[no] ip ospf authentication [null | message-digest] [if-ip-addr]
[no] ip ospf authentication-key key [if-ip-addr]
[no] ip ospf message-digest-key key-id md5 key [if-ip-addr]
```

- *if-ip-addr* — Interface IPv4 address
- *key* — Authentication key password (up to 8 or 16 characters)
- *key-id* — MD5 authentication key Identifier (1 to 255)
- *if-ip-addr* — Interface IP address

#### Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf authentication
sonic(conf-if-Eth1/1)# ip ospf authentication-key ospfpswd

sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# ip ospf authentication 10.10.3.2
sonic(conf-if-Eth1/2)# ip ospf authentication-key ospfpswd 10.10.3.2

sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# ip ospf authentication message-digest
sonic(conf-if-Eth1/3)# ip ospf message-digest-key 1 md5 ospfpswd1
sonic(conf-if-Eth1/3)# ip ospf message-digest-key 2 md5 ospfpswd2
sonic(conf-if-Eth1/3)# ip ospf message-digest-key 9 md5 ospfpswd9

sonic(config)# interface Eth1/4
sonic(conf-if-Eth1/4)# ip ospf authentication null
```

Configure the OSPFv2 interface session BFD under OSPF interface configuration mode. OSPF interface BFD is disabled by default.

```
[no] ip ospf bfd
```

#### Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf bfd
sonic(conf-if-Eth1/1)#
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# ip ospf network broadcast
```

Configure the OSPFv2 interface cost under OSPF interface configuration mode. The cost value is set to router-LSA's metric field and used for SPF calculation.

```
[no] ip ospf cost cost-value [if-ip-addr]
```

- *if-ip-addr* — Interface IP address
- *cost-value* — Interface cost (1 to 65535)

#### Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf cost 50
```

Configure OSPFv2 interface session MTU ignore under OSPF interface configuration mode. MTU is not ignored by default. MTU values of OSPF session interface ends shall match if MTU ignore is nor configured. Configuring MTU ignore does not guarantee the session establishment when there are large number of OSPF routes prefixes are present and router LSA message cannot accommodate them within the MTU size.

```
[no] ip ospf mtu-ignore [if-ip-addr]
```

- *if-ip-addr* — Interface IP address

#### Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf mtu-ignore
```

Configure the OSPFv2 interface session hello timers under OSPF interface configuration mode. When configured, Hello packet is sent every timer value seconds on the specified interface. This value must be the same for all routers attached to a common network. The default value is 10 seconds.

```
[no] ip ospf hello-interval time-interval [if-ip-addr]
```

- *if-ip-addr* — Interface IP address
- *time-interval* — Timer values (1 to 65535 seconds)

#### Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf hello-interval 15
```

Configure the OSPFv2 interface session transmit delay timers under OSPF interface configuration mode. When configured, LSAs' age should be incremented by this value when transmitting. The default value is 1 second.

```
[no] ip ospf transmit-delay time-interval [if-ip-addr]
```

- *if-ip-addr* — Interface IP address
- *time-interval* — Timer values (1 to 65535 seconds)

## Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf transmit-delay 20
```

Configure OSPFv2 interface session retransmit interval timers under OSPF interface configuration mode. This value is used when retransmitting Database Description and Link State Request packets. The default value is 5 seconds.

```
[no] ip ospf retransmit-interval time-interval [if-ip-addr]
```

- *if-ip-addr* — Interface IP address
- *time-interval* — Timer values (3 to 65535 seconds)

## Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf retransmit-interval 25
```

Configure the OSPFv2 interface dead interval under OSPF interface configuration mode.

OSPF Router Dead Interval timer value is used for session wait or inactivity Timer. This value must be the same for all routers attached to a common network. The default value is 40 seconds. If 'minimal' is specified instead of explicit wait time, then the dead-interval is set to 1 second and one must specify a hello-multiplier. The hello-multiplier specifies how many Hellos to send per second. The multiplier value can be from 1 (every 500ms) to 10 (every 50ms). Thus one can have 1s convergence time for OSPF. If this form is specified, then the hello-interval advertised in Hello packets is set to 0 and the hello-interval on received Hello packets is not checked, thus the hello-multiplier need NOT be the same across multiple routers on a common link.

```
[no] ip ospf dead-interval time-interval [if-ip-addr]
[no] ip ospf dead-interval minimal hello-multiplier multiplier [if-ip-addr]
```

- *if-ip-addr* — Interface IP address
- *time-interval* — Timer values (1 to 65535 seconds)
- *multiplier* — Dead interval hello multiplier value (1 to 10 seconds)

## Example

```
sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# ip ospf dead-interval 60

sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# ip ospf dead-interval minimal hello-multiplier 4
```

# OSPF operational data display commands

SONiC provides display command to show the operational status of OSPF router, OSPF sessions and OSPF interfaces.

## View the OSPF router information

```
show ip ospf [vrf vrf-name]
```

- *vrf-name* — VRF name

```
sonic# show ip ospf

OSPF Routing Process, Router ID: 1.1.1.1
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is enabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 0 millisecond(s)
Minimum hold time between consecutive SPFs 50 millisecond(s)
Maximum hold time between consecutive SPFs 5000 millisecond(s)
Hold time multiplier is currently 1
time is 92031756
SPF algorithm last executed 1065d4h22m ago
Last SPF duration 0.0s
```

```

SPF timer is inactive
LSA minimum interval 5000 msecs
LSA minimum arrival 1000 msecs
Write Multiplier set to 20
Refresh timer 10 secs
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of areas attached to this router: 2
Area ID: 0.0.0.0 (Backbone)
 Number of interfaces in this area: Total: 1 , Active: 1
 Number of fully adjacent neighbors in this area: 1
 Area has no authentication
 SPF algorithm executed 8 times
 Number of LSA 3
 Number of router LSA 2. Checksum Sum 0x40f64b4000000000
 Number of network LSA 1. Checksum Sum 0x40d5adc000000000
 Number of summary LSA 0. Checksum Sum 0x0
 Number of ASBR summary LSA 0. Checksum Sum 0x0
 Number of NSSA LSA 0. Checksum Sum 0x0
 Number of opaque link LSA . Checksum Sum 0x
 Number of opaque area LSA 0. Checksum Sum 0x0
Area ID: 0.0.0.1
 Number of interfaces in this area: Total: 1 , Active: 1
 Number of fully adjacent neighbors in this area: 0
 Area has no authentication
 SPF algorithm executed 1 times
 Number of LSA 2
 Number of router LSA 0. Checksum Sum 0x0
 Number of network LSA 0. Checksum Sum 0x0
 Number of summary LSA 2. Checksum Sum 0x40f1f61000000000
 Number of ASBR summary LSA 0. Checksum Sum 0x0
 Number of NSSA LSA 0. Checksum Sum 0x0
 Number of opaque link LSA . Checksum Sum 0x
 Number of opaque area LSA 0. Checksum Sum 0x0

```

### **View the OSPF neighbor information**

```
show ip ospf [vrf vrf-name] neighbor [detail | if-name | nbr-ip]
```

- *vrf-name* — VRF name
- *if-name* — OSPF interface name
- *nbr-ip* — Neighbor router ID

```

sonic# show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface RXmtL
RqstL DBsmL
10.59.142.247 1 Full/Backup 37.343s 64.1.1.2 Eth 1/2:64.1.1.1 0
0 0

sonic# show ip ospf neighbor Eth1/3 | no-more

Neighbor ID Pri State Dead Time Address Interface RXmtL
RqstL DBsmL
2.2.2.2 1 Full/Backup 38.245s 64.1.1.2 Eth 1/4:64.1.1.1 0
0 0

sonic# show ip ospf neighbor detail
Neighbor 10.59.142.247, interface address 64.1.1.2
In the area 0.0.0.0 via interface Eth 1/2
Neighbor priority is 1, State is Full, 6 state changes
Most recent state change statistics:
 Progressive change 7h3m25s ago
 DR is 64.1.1.1, BDR is 64.1.1.2
 Options 2 *|-|-|-|-|E|-|
 Dead timer due in 30.687s
 Database Summary List 0
 Link State Request List 0
 Link State Retransmission List 0
 Thread Inactivity Timer on
 Thread Database Description Retransmission off

```

```

Thread Link State Request Retransmission on
Thread Link State Update Retransmission on

Leaf1# show ip ospf neighbor 2.2.2.2
Neighbor 2.2.2.2, interface address 64.1.1.2
 In the area 0.0.0.0 via interface Eth 1/2
 Neighbor priority is 1, State is Full, 5 state changes
 Most recent state change statistics:
 Progressive change 0h1m11s ago
 DR is 64.1.1.1, BDR is 64.1.1.2
 Options 2 *|-|-|-|-|E|-|
 Dead timer due in 33.203s
 Database Summary List 0
 Link State Request List 0
 Link State Retransmission List 0
 Thread Inactivity Timer on
 Thread Database Description Retransmission off
 Thread Link State Request Retransmission on
 Thread Link State Update Retransmission on

```

### View the OSPF interface information

```
show ip ospf interface [if-name | traffic]
```

- *if-name* — OSPF interface name

```

sonic# show ip ospf interface
VRF Name: default
Eth 1/2 is up
 ifindex 128, MTU 9100 bytes, BW 25000 Mbit UP,BROADCAST,RUNNING,MULTICAST
 Internet Address 64.1.1.1/24, Broadcast 64.1.1.255, Area 0.0.0.0
 MTU mismatch detection: enabled
 Router ID 10.59.143.131, Network Type BROADCAST, Cost: 4
 Transmit Delay is 1 sec, State DR, Priority 1
 Backup Designated Router (ID) 10.59.142.247, Interface Address 64.1.1.2
 Saved Network-LSA sequence number 0x8000000f
 Multicast group memberships: OSPFAllRouters OSPFDesignatedRoute
 Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
 Hello due in 9.023s
 Neighbor Count is 1, Adjacent neighbor count is 1

```

```

sonic# show ip ospf interface Eth 1/3
VRF Name: default
Eth 1/4 is up
 ifindex 926, MTU 9100 bytes, BW 25000 Mbit UP,BROADCAST,RUNNING,MULTICAST
 Internet Address 65.1.1.1/24, Broadcast 65.1.1.255, Area 0.0.0.1
 MTU mismatch detection: enabled
 Router ID 1.1.1.1, Network Type BROADCAST, Cost: 4
 Transmit Delay is 1 sec, State DR, Priority 1
 Backup Designated Router (ID) 2.2.2.2, Interface Address 65.1.1.2
 Multicast group memberships: OSPFAllRouters OSPFDesignatedRoute
 Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
 Hello due in 7.957s
 Neighbor Count is 1, Adjacent neighbor count is 1

```

```
sonic# show ip ospf interface traffic
```

| Interface | HELLO<br>Rx/Tx | DB-Desc<br>Rx/Tx | LS-Req<br>Rx/Tx | LS-Update<br>Rx/Tx | LS-Ack<br>Rx/Tx |
|-----------|----------------|------------------|-----------------|--------------------|-----------------|
| Eth 1/2   | 2563/2563      | 3/3              | 1/1             | 17/30              | 29/16           |

```
Leaf1# show ip ospf interface traffic Eth1/3
```

| Interface | HELLO<br>Rx/Tx | DB-Desc<br>Rx/Tx | LS-Req<br>Rx/Tx | LS-Update<br>Rx/Tx | LS-Ack<br>Rx/Tx |
|-----------|----------------|------------------|-----------------|--------------------|-----------------|
| Eth1/3    | 19/22          | 2/3              | 1/1             | 3/3                | 2/2             |

## View the OSPF Database information

```
show ip ospf [vrf vrf-name] database [asbr-summary | external | network | router | summary | opaque-link]
```

- *vrf-name* — VRF name

```
sonic# show ip ospf database
VRF Name: default

 OSPF Router with ID (5.5.5.5)

 Router Link States (Area 0.0.0.0)

Link ID ADV Router Age Seq# CkSum Link count
3.3.3.3 3.3.3.3 988 0x80000003 0x04ec 1
5.5.5.5 5.5.5.5 988 0x80000008 0x6f6b 1

 Net Link States (Area 0.0.0.0)

Link ID ADV Router Age Seq# CkSum
10.10.10.2 5.5.5.5 988 0x80000002 0xcc38

 Link-Local Opaque-LSA (Area 0.0.0.0)

Opaque-Type/Id ADV Router Age Seq# CkSum
3.0.0.0 3.3.3.3 89 0x80000001 0x4a24
```

```
sonic# show ip ospf database network
VRF Name: default

 OSPF Router with ID (10.59.143.131)

 Net Link States (Area 0.0.0.0)

LS age: 1602
Options: 0x2 : *|-|-|-|-|E|-
LS Flags: 0x3
LS Type: network-LSA
Link State ID: 64.1.1.1 (address of Designated Router)
Advertising Router: 10.59.143.131
LS Seq Number: 8000000f
Checksum: 0x1c70
Length: 32

Network Mask: /24
Attached Router: 10.59.142.247

Attached Router: 10.59.143.131
```

```
sonic# show ip ospf database summary
VRF Name: default

 OSPF Router with ID (1.1.1.1)

 Summary Link States (Area 0.0.0.0)

LS age: 468
Options: 0x2 : *|-|-|-|-|E|-
LS Flags: 0x11
LS Type: summary-LSA
Link State ID: 65.1.1.0 (summary Network Number)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0x0e04
Length: 28

Network Mask: /24
TOS: 0 Metric: 4
```

```

LS age: 429
Options: 0x2 : *|-|-|-| -|E|-|
LS Flags: 0x6
LS Type: summary-LSA
Link State ID: 65.1.1.0 (summary Network Number)
Advertising Router: 2.2.2.2
LS Seq Number: 80000002
Checksum: 0xed1f
Length: 28

Network Mask: /24
TOS: 0 Metric: 4

Summary Link States (Area 0.0.0.1)

LS age: 468
Options: 0x2 : *|-|-|-| -|E|-|
LS Flags: 0x11
LS Type: summary-LSA
Link State ID: 64.1.1.0 (summary Network Number)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0x1bf7
Length: 28

Network Mask: /24
TOS: 0 Metric: 4

LS age: 429
Options: 0x2 : *|-|-|-| -|E|-|
LS Flags: 0x6
LS Type: summary-LSA
Link State ID: 64.1.1.0 (summary Network Number)
Advertising Router: 2.2.2.2
LS Seq Number: 80000002
Checksum: 0xfa13
Length: 28

Network Mask: /24
TOS: 0 Metric: 4

```

```

sonic# show ip ospf database asbr-summary
VRF Name: default

OSPF Router with ID (1.1.1.1)

ASBR-Summary Link States (Area 0.0.0.0)

LS age: 38
Options: 0x2 : *|-|-|-| -|E|-|
LS Type: summary-LSA
Link State ID: 2.2.2.2 (AS Boundary Router address)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0xb41
Length: 28

Network Mask: /0
TOS: 0 Metric: 4

```

```

sonic# show ip ospf database external
VRF Name: default

OSPF Router with ID (1.1.1.1)

AS External Link States

```

```

LS age: 52
Options: 0x2 : *|-|-|-|-|E|-
LS Flags: 0x6
LS Type: AS-external-LSA
Link State ID: 25.1.1.1 (External Network Number)
Advertising Router: 2.2.2.2
LS Seq Number: 80000001
Checksum: 0x0892
Length: 36

Network Mask: /32
Metric Type: 2 (Larger than any link state path)
TOS: 0
Metric: 20
Forward Address: 0.0.0.0
External Route Tag: 0

```

```

sonic# show ip ospf database self-originate
VRF Name: default

OSPF Router with ID (1.1.1.1)

 Router Link States (Area 0.0.0.0)

Link ID ADV Router Age Seq# CkSum Link count
1.1.1.1 1.1.1.1 777 0x80000004 0x7b42 1

 Net Link States (Area 0.0.0.0)

Link ID ADV Router Age Seq# CkSum
64.1.1.1 1.1.1.1 777 0x80000001 0x8581

 Summary Link States (Area 0.0.0.0)

Link ID ADV Router Age Seq# CkSum Route
65.1.1.0 1.1.1.1 816 0x80000001 0x0e04 65.1.1.0/24

 ASBR-Summary Link States (Area 0.0.0.0)

Link ID ADV Router Age Seq# CkSum
2.2.2.2 1.1.1.1 360 0x80000001 0xb41

 Router Link States (Area 0.0.0.1)

Link ID ADV Router Age Seq# CkSum Link count
1.1.1.1 1.1.1.1 776 0x80000004 0x8d2e 1

 Net Link States (Area 0.0.0.1)

Link ID ADV Router Age Seq# CkSum
65.1.1.1 1.1.1.1 776 0x80000001 0x788d

 Summary Link States (Area 0.0.0.1)

Link ID ADV Router Age Seq# CkSum Route
64.1.1.0 1.1.1.1 816 0x80000001 0x1bf7 64.1.1.0/24

```

```

sonic# show ip ospf database network adv-router 1.1.1.1
VRF Name: default

OSPF Router with ID (1.1.1.1)

 Net Link States (Area 0.0.0.0)

LS age: 886
Options: 0x2 : *|-|-|-|-|E|-
LS Flags: 0x3
LS Type: network-LSA
Link State ID: 64.1.1.1 (address of Designated Router)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001

```

```

Checksum: 0x8581
Length: 32

Network Mask: /24
Attached Router: 1.1.1.1

Attached Router: 2.2.2.2

Net Link States (Area 0.0.0.1)

LS age: 886
Options: 0x2 : *|-|-|-|E|-
LS Flags: 0x3
LS Type: network-LSA
Link State ID: 65.1.1.1 (address of Designated Router)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0x788d
Length: 32

Network Mask: /24
Attached Router: 1.1.1.1

Attached Router: 2.2.2.2

```

```

sonic# show ip ospf database opaque-link
VRF Name: default

OSPF Router with ID (5.5.5.5)

Link-Local Opaque-LSA (Area 0.0.0.0)

LS age: 94
Options: 0x66 : *|O|-|-|-|E|-
LS Flags: 0x6
LS Type: Link-Local Opaque-LSA
Link State ID: 3.0.0.0 (Link-Local Opaque-Type/ID)
Advertising Router: 3.3.3.3
LS Seq Number: 80000001
Checksum: 0x4a24
Length: 44

```

### **View the OSPF route information**

```
show ip ospf [vrf vrf-name] route
```

- *vrf-name* — VRF name

```

sonic# show ip ospf vrf Vrf1 route | no-more
VRF Name: Vrf1
===== OSPF network routing table =====
N 101.1.1.0/24 [10] area: 0.0.0.0
 directly attached to Vlan101

===== OSPF router routing table =====

===== OSPF external routing table =====

```

## **OSPFv2 graceful restart**

When routers that are participating in OSPF are restarted, there is a period of traffic loss until the routers come back online and relearn the routes.

RFC 3623 specifies the Graceful Restart enhancement to OSPF as follows:

- The router attempting a graceful restart originates link-local Opaque-LSAs (Grace-LSAs), announcing its intention to perform a graceful restart within a specified amount of time or grace period.

- During the grace period, the neighbors continue to announce the restarting router in their LSAs as if it were fully adjacent (that is, OSPF neighbor state Full), but only if the network topology remains static (that is, the contents of the LSAs in the link-state database having LS types 1 to 5, and 7 remain unchanged, and periodic refreshes are allowed).

Graceful restart allows the restarting router to inform its neighbors that it is going to restart. As the neighbors are informed of the condition, they continue forwarding traffic to the restarting node. As the forwarding table of the restarting node is preserved during graceful restart, traffic loss is avoided.

## Planned outages

Enterprise SONiC supports OSPF graceful restart only for planned outages.

Graceful restart is enabled for OSPF instances. System warm restart is triggered using the `warm-reboot` command.

## Restarting and helper nodes

With graceful restart, there are two types of devices that participate in the OSPF network. They are restarting and helper nodes.

The restarting node is the device that gracefully restarts without having a traffic loss. When the restarting node performs a graceful restart, it floods link-local opaque LSAs ( grace-LSAs) on all OSPF interfaces. These grace LSAs inform the helper router that the neighbor intends to restart.

Helper nodes help the restarting node so that there is no traffic loss. A helper node monitors the network for topology changes. As long as there is no change in the network, the helper continues to advertise its LSAs as if the restarting node had remained in continuous OSPF operation. The helper LSAs continue to list an adjacency to the restarting node over network segment, regardless of the synchronization state of the restarting node.

## Configure OSPFv2 graceful restart

To configure graceful restart, follow these steps:

1. Enable graceful restart.

```
sonic(config-router-ospf)# graceful-restart [grace-period grace-period]
```

- *grace-period* — The grace period before which the neighbors or helpers deem the restarting node dead. The range is from 1 to 1800 seconds. The default grace period is 120 seconds

2. Configure OSPFv2 opaque capability to help the restarting router to initiate grace LSAs.

```
sonic(config-router-ospf)# capability opaque
```

3. Configure OSPFv2 graceful restart helper on the system for all or a specific router ID. Do this on all nodes that you are using to help the restarting router.

```
sonic(config-router-ospf)# graceful-restart helper enable [router-id]
```

- *router-id* — Configure graceful restart helper support for a specific neighbor using the router ID.

4. (Optional) Configure the grace time on the helper node. This configuration determines the time period for the helper to support graceful restart.

```
sonic(config-router-ospf)# graceful-restart helper supported-grace-time grace-time
```

- *grace-time* — Configure the grace time. The range is from 10 to 1800 seconds. The default value is 120 seconds.

5. (Optional) Configure strict LSA checking on the helper node. If this command is configured, the helper cancels graceful restart when an LSA change occurs, which affects the restarting router. By default, strict LSA checking is enabled.

```
sonic(config-router-ospf)# graceful-restart helper strict-lsa-checking
```

6. (Optional) Configure helper support for only planned restarts.

```
sonic(config-router-ospf)# graceful-restart helper planned-only
```

## View OSPFv2 graceful restart information

Use the following commands to view graceful restart helper information.

View general OSPD information including if graceful restart and opaque capability are enabled.

```
sonic# show ip ospf
VRF Name: default
OSPF Routing Process, Router ID: 1.1.1.2
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is disabled
OpaqueCapability flag is enabled
Graceful-Restart is enabled
Stub router advertisement is configured
 Enabled for 600s after start-up
Initial SPF scheduling delay 0 millisec(s)
Minimum hold time between consecutive SPFs 50 millisec(s)
Maximum hold time between consecutive SPFs 5000 millisec(s)
Hold time multiplier is currently 1
SPF algorithm last executed 19h39m37s ago
Last SPF duration 72320 usecs
SPF timer is inactive
LSA minimum interval 5000 msec
LSA minimum arrival 0 msec
Write Multiplier set to 20
Refresh timer 10 secs
Maximum multiple paths(ECMP) supported 256
Number of external LSA 0. Checksum Sum 0x00000000
Number of opaque AS LSA 0. Checksum Sum 0x00000000
Number of areas attached to this router: 1
Area ID: 0.0.0.0 (Backbone)
 Number of interfaces in this area: Total: 224 , Active: 224
 Number of fully adjacent neighbors in this area: 32
 Area has simple password authentication
 SPF algorithm executed 247 times
 Number of LSA 6068
 Number of router LSA 6. Checksum Sum 0x00020692
 Number of network LSA 62. Checksum Sum 0x001ab353
 Number of summary LSA 6000. Checksum Sum 0x0bc36229
 Number of ASBR summary LSA 0. Checksum Sum 0x00000000
 Number of NSSA LSA 0. Checksum Sum 0x00000000
 Number of opaque link LSA 0. Checksum Sum 0x00000000
 Number of opaque area LSA 0. Checksum Sum 0x00000000
```

View the OSPF neighbor information and details about graceful restart helper.

```
sonic# show ip ospf neighbor detail
Neighbor 13.13.13.13, interface address 192.168.10.1
 In the area 0.0.0.0 via interface Ethernet64
 Neighbor priority is 1, State is Full, 6 state changes
 Most recent state change statistics:
 Progressive change 17h32m19s ago
 DR is 192.168.10.1, BDR is 192.168.10.2
 Options 66 *|O|-|-|-|-|E|-|
 Dead timer due in 0.717s
 Database Summary List 0
 Link State Request List 0
 Link State Retransmission List 0
 Thread Inactivity Timer on
 Thread Database Description Retransmission off
 Thread Link State Request Retransmission on
 Thread Link State Update Retransmission on
 Graceful restart Helper info:
 Graceful Restart HELPER Status: Inprogress
 Graceful Restart grace period time: 250 (seconds).
 Graceful Restart reason: Software restart
```

View OSPF graceful restart helper information.

```
sonic# show ip ospf graceful-restart helper
VRF Name: default
OSPF Router with ID (14.14.14.14)
Graceful restart helper support enabled.
Strict LSA check is enabled.
Helper supported for planned restarts only.
Supported Graceful restart interval: 1600(in seconds).
Enable Router List:
['13.13.13.13']
```

View OSPF detailed graceful restart helper information.

```
sonic# show ip ospf graceful-restart helper detail
VRF Name: default
OSPF Router with ID (14.14.14.14)
Graceful restart helper support enabled.
Strict LSA check is enabled.
Helper supported for planned restarts only.
Supported Graceful restart interval: 1600(in seconds).
Enable Router List:
['13.13.13.13']
Number of Active neighbours in graceful restart: 4
Neighbour 1:
Address: 192.168.10.1
Routerid: 13.13.13.13
Received Grace period: 250(in seconds).
Actual Grace period: 250(in seconds).
Remaining GraceTime: 245(in seconds).
Graceful Restart reason: Software restart.
Neighbour 2:
Address: 192.168.20.1
Routerid: 13.13.13.13
Received Grace period: 250(in seconds).
Actual Grace period: 250(in seconds).
Remaining GraceTime: 245(in seconds).
Graceful Restart reason: Software restart.
Neighbour 3:
Address: 192.168.30.1
Routerid: 13.13.13.13
Received Grace period: 250(in seconds).
Actual Grace period: 250(in seconds).
Remaining GraceTime: 245(in seconds).
Graceful Restart reason: Software restart.
Neighbour 4:
Address: 192.168.40.1
Routerid: 13.13.13.13
Received Grace period: 250(in seconds).
Actual Grace period: 250(in seconds).
Remaining GraceTime: 245(in seconds).
Graceful Restart reason: Software restart.
```

## Route-maps

Route-maps are used when distributing routes into an OSPF or BGP routing process. They are also used when generating a default route into an OSPF routing process.

A route-map defines which of the routes from the specified routing protocol are allowed to be redistributed into the target routing process. Route-maps have many features in common with widely known ACLs.

Common traits between route-maps and ACLs:

- They are an ordered sequence of individual statements, and each has a permit and deny result. Evaluation of an ACL or a route-map consists of a list scan, in a predetermined order, and an evaluation of the criteria in each statement that matches. A list scan is removed once the first statement match is found, and an action that is associated with the statement match is performed.
- They are generic mechanisms. Criteria matches and match interpretation are dictated by the way that they are applied, and the feature that uses them. The same route-map applied to different features may be interpreted differently.

Differences between route-maps and ACLs:

- Route-maps are more flexible than ACLs and can verify routes based on criteria which ACLs cannot verify. For example, a route-map can verify if the type of route is internal.
- Each ACL ends with an implicit deny statement. If the end of a route-map is reached during match attempts, the result depends on the specific application of the route-map. Route-maps that are applied to *redistribution* behave the same way as ACLs — if the route does not match any statement in a route-map, then the route redistribution is denied as if the route-map contained a deny statement at the end.

## Permit and deny statements

Route-maps can have permit and deny statements. The deny statement rejects route matches from redistribution. You can use an ACL as the matching criteria in the route-map. Because ACLs also have permit and deny statements, these rules apply when packet matches the ACL:

- ACL permit plus route-map permit — routes are redistributed
- ACL permit plus route-map deny — routes are not redistributed
- ACL deny plus route-map permit or deny — the route-map statement is not matched, and the next route-map statement is evaluated

## Match and set statement values

Each route-map statement has two types of values:

- A match value selects routes to which this statement should be applied
- A set value modifies information that is to be redistributed into the target protocol

For each route that is redistributed, the router first evaluates the match criteria of a statement in the route-map. If the match criteria succeeds, then the route is redistributed or rejected as dictated by the permit or deny statement, and some of its attributes may be modified by the values set from the set commands.

If the match criteria fails, then this statement is not applicable to the route, and the software goes to evaluate the route against the next statement in the route-map. Scanning of the route-map continues until a statement is found that matches the route or until the end of the route-map is reached.

A match or set value in each statement can be missed or repeated several times, if one of these conditions exist:

- If several match entries are present in a statement, all must succeed for a given route in order for that route to match the statement (a logical AND algorithm is applied)
- If a match entry sees several objects in one entry, either of them should match (a logical OR algorithm is applied)
- If a match entry is not present, all routes match the statement
- If a set entry is not present in the route-map permit statement, then the route is redistributed without modification of its current attributes

A route-map statement without a match or set entry does perform an action. An empty permit statement allows a redistribution of the remaining routes without modification. An empty deny statement does not allow a redistribution of other routes as this is the default action if a route-map is completed scanned but no explicit match is found.

## Create a route-map

Create route-maps to filter routes using permit and deny statements. Configure match statements to select routes, and configure set statements to modify the BGP attributes in a matching route.

Apply a route-map filter to the address families of BGP neighbors and BGP peer-groups in the inbound or outbound direction using `route-map map-name {in | out}`.

1. Create a route-map to match the route parameters listed in the next step. Specify a permit or deny statement to configure how matching routes are handled. Enter the sequence number for the order in which the statement is processed in the map.

```
sonic(config)# route-map map-name {permit | deny} sequence-number
```

2. In route-map configuration mode, enter any of these match statements to select routes.

```
sonic(config-route-map)# match as-path acl-name
sonic(config-route-map)# match community community-list-name
```

```

sonic(conf-route-map) # match ext-community extcommunity-list-name
sonic(conf-route-map) # match interface interface
sonic(conf-route-map) # match ip address prefix-list prefix-list-name
sonic(conf-route-map) # match ipv6 address prefix-list prefix-list-name
sonic(conf-route-map) # match metric value
sonic(conf-route-map) # match route-type {internal | external}
sonic(conf-route-map) # match origin {egp | igrp | incomplete}
sonic(conf-route-map) # match tag value
sonic(conf-route-map) # match local-preference value
sonic(conf-route-map) # match peer ip-address
sonic(conf-route-map) # match ip next-hop prefix-list prefix-list-name
sonic(conf-route-map) # call route-map-name
sonic(conf-route-map) # match source-protocol {bgp | ospf | static | connected}

```

3. In route-map configuration mode, enter any of these set statements to change the specified BGP attribute in matching routes.

```

sonic(conf-route-map) # set as-path prepend list
sonic(conf-route-map) # set community options
sonic(conf-route-map) # set ext-community options
sonic(conf-route-map) # set comm-list community-list-name delete
sonic(conf-route-map) # set ip next-hop number
sonic(conf-route-map) # set ipv6 next-hop number
sonic(conf-route-map) # set local-preference value
sonic(conf-route-map) # set metric value
sonic(conf-route-map) # set origin {igrp | egp | incomplete}

```

### Configure route-map example

```

sonic(config)# route-map map1 permit 10
sonic(conf-route-map) # match as-path ASlist1
sonic(conf-route-map) # match ext-community comm3
sonic(conf-route-map) # match interface Eth1/2
sonic(conf-route-map) # set metric 100
sonic(conf-route-map) # set origin egp
sonic(conf-route-map) # set local-preference 10000

```

To remove a configured value in a route-map entry, enter the no version of the match or set command.

```

sonic(config)# route-map map1 permit 10
sonic(conf-route-map) # no match as-path
sonic(conf-route-map) # no set origin

```

## Static routes

Configure fixed, static routes to ensure that routed traffic can be exchanged with a specified destination device. For example, use a static route as a backup in case a dynamic route is not available, or to reach a network gateway if no other route is available.

Static and dynamic routes can co-exist in the routing table. You can also use static routes to switch between two networks to transfer routing information between routing protocols. You configure static routes for management and non-management traffic.

**(i)** **NOTE:** Configuring a static IP route to blackhole traffic sent from the Management interface (Management0) takes effect only if the Management VRF is configured - see [Virtual routing and forwarding](#).

### Configure an IPv4 or IPv6 static route

```

sonic(config)# ip route [vrf vrf-name] dest-ip-prefix {next-hop-ip [interface interface-type interface-number] | interface interface-type interface-number | blackhole} [nexthop-vrf vrf-name] [tag tag-name] [track id] [dest-metric]

```

```

sonic(config)# ipv6 route [vrf vrf-name] dest-ipv6-prefix {next-hop-ipv6 [interface interface-type interface-number] | interface interface-type interface-number | blackhole} [nexthop-vrf vrf-name] [tag tag-name] [track id] [dest-metric]

```

To configure a static IPv4 or IPv6 route, enter the destination address prefix and these optional values:

- (Optional) `vrf vrf-name` — Enter the name of the VRF instance in which you want to configure a static route. For management VRF, use `mgmt`.
- `dest-prefix` — Enter the IPv4/IPv6 prefix of the destination device. Enter an IPv4 prefix in the format `A.B.C.D/mask`, where `mask` is an IPv4 prefix-mask number from 1 to 32; for example, `10.10.10.0/24`. Enter an IPv6 prefix in the format `x:x:x:x:x:x:x:/mask`, where `mask` is an IPv6 prefix-mask number is from 1 to 128; for example, `2001:db8:1234:0000::/64`.
- (Optional) `next-hop-ip` — Enter a next-hop IPv4/IPv6 address as the gateway for the destination prefix.
- (Optional) `interface interface-type interface-number` — Specify the switch interface through which statically routed IPv4/IPv6 traffic passes to the destination, where `interface-type interface-number` is one of these values:
  - `interface Eth slot/port [/breakout-port]`
  - `interface PortChannel portchannel-number`
  - `interface Vlan vlan-id`
  - `interface Loopback number`
  - `interface Management 0`
- (Optional) `blackhole` — Blocks traffic from a suspected source or a denial of service (DoS) attack by dynamically routing the traffic to a destination device or a data collection device.
- (Optional) `nexthop-vrf vrf-name` — Enter the name of the VRF instance used on the next-hop device if the previously specified VRF or interface is in a different VRF. For management VRF, use `mgmt`. Use this parameter to configure static route leaking.
- (Optional) `tag tag-name` — Enter a tag number to use to match the route in route maps, from 1 to 4294967295.
- (Optional) `track id` — Enter an ID number to track the route to evaluate performance in Service Level Agreements (SLAs), from 1 to 255.
- (Optional) `dest-metric` — Enter the number that determines the precedence of routing paths, from 1 to 255 — the lower the number, the higher the route precedence. The static route default is 1.

#### Configure IPv4 static route

```
sonic(config)# ip route vrf Vrf-RED 7.7.7.0/24 interface Eth1/4 nexthop-vrf Vrf-GREEN 36
sonic(config)# ip route 4.4.4.0/24 4.4.4.1 nexthop-vrf Vrf-GREEN 200
sonic(config)# ip route vrf Vrf-RED 3.3.3.0/24 interface Eth1/2 nexthop-vrf Vrf-GREEN 150
```

To remove an IPv4 static route, enter the no version of the command without the distance metric:

```
sonic(config)# no ip route vrf Vrf-RED 7.7.7.0/24 interface Eth1/4 nexthop-vrf Vrf-GREEN
sonic(config)# no ip route 4.4.4.0/24 4.4.4.1 nexthop-vrf Vrf-GREEN
sonic(config)# no ip route vrf Vrf-RED 3.3.3.0/24 interface Eth1/2 nexthop-vrf Vrf-GREEN
```

#### Configure IPv4 static route through management interface

Configure an IPv4 static route through management interface when the management VRF is not configured on the system.

```
sonic(config)# ip route 10.5.6.6/24 interface Management0
```

#### Configure IPv4 static route and leak the route to management VRF

```
sonic(config)# ip route vrf Vrf-common 10.1.1.1/32 interface Loopback99 nexthop-vrf mgmt
```

#### Configure IPv6 static route

```
sonic(config)# ipv6 route 3030::3300/120 3030::3301 36
sonic(config)# ipv6 route 2001:db6::/32 interface Eth1/2 150
sonic(config)# ipv6 route 2006::/24 blackhole 200
```

To remove an IPv6 static route, enter the no version of the command without the distance metric:

```
sonic(config)# no ipv6 route 3030::3300/120 3030::3301
sonic(config)# no ipv6 route 2001:db6::/32 interface Eth1/2
sonic(config)# no ipv6 route 2006::/24 blackhole
```

#### Configure IPv6 static route through management interface

Configure an IPv6 static route through management interface when the management VRF is not configured on the system.

```
sonic(config)# ipv6 route 2020:FF21:1:1::/64 interface Management0
```

### Configure IPv6 static route and leak the route to management VRF

```
sonic(config)# ip route vrf Vrf-common 2020:FF21:1:1::/64 interface Loopback99 nexthop-vrf mgmt
```

### View static routes

```
sonic# show ip route static
Codes: K - kernel route, C - connected, S - static, B - BGP, O - OSPF
 > - selected route, * - FIB route, q - queued route, r - rejected route,
 # - not installed in hardware
 Destination Gateway Dist/Metric Last Update

S# 3.3.3.0/24 Direct Eth1/2 150/0 00:33:31 ago
S# 6.6.6.0/24 via 6.6.6.1 Eth1/1 1/0 00:00:03 ago
S# 7.7.7.0/24 via 7.7.7.1 Eth1/4 36/0 00:01:56 ago
```

```
sonic# show ipv6 route static
Codes: K - kernel route, C - connected, S - static, B - BGP, O - OSPF
 > - selected route, * - FIB route, q - queued route, r - rejected route,
 # - not installed in hardware
 Destination Gateway Dist/Metric Last Update

S# 2001:db5::/32 Direct Eth1/3 1/0 00:35:57 ago
S# 3020::/64 via 7070::7070 Eth1/1 1/0 00:00:41 ago
S# 3030::3300/120 via 3030::3301 Eth1/4 36/0 00:36:44 ago
```

```
sonic# show ip route vrf mgmt static
Codes: K - kernel route, C - connected, S - static, B - BGP, O - OSPF
 > - selected route, * - FIB route, q - queued route, r - rejected route, # - not
 installed in hardware
 Destination Gateway Dist/Metric Last Update

S 100.0.0.0/24 Direct Vlan100(vrf Vrf-common) 1/0 00:03:10 ago
```

=

=

```
sonic# show ipv6 route vrf mgmt static
Codes: K - kernel route, C - connected, S - static, B - BGP, O - OSPF
 > - selected route, * - FIB route, q - queued route, r - rejected route, # -
 not installed in hardware
 Destination Gateway Dist/Metric Last Update

S 2001:db5::/32 Direct Eth1/17 1/0 00:11:26 ago
S 2001:db8::/32 via 3030::3302 Vlan100(vrf Vrf-common) 200/0 00:11:38 ago
```

### Static routes in gNMI and REST API requests

If you use gNMI remote calls or REST API HTTP requests to retrieve or configure static routes, you must specify the `index` key in the OpenConfig YANG data model. This key is used to uniquely identify a next-hop entry. Enter the `index` key in the format:

```
[interface_type]_[next-hop-ip]_[vrf-name]
```

Enter the interface, next-hop, and next-hop VRF values only if they are configured for the next-hop device. Separate the values with an underscore (\_).The `index` key is used to define the next-hop device in non-blackhole routes. For blackhole routes, enter `DROP` as the `index` value. To specify the management interface, use `Management0`.

### REST API

```
Eth1%2F1_2.2.2.2_default
3.3.3.3_Vrf-RED
DROP
```

## gNMI

```
Eth1/1 2.2.2.2 default
3.3.3.3 Vrf-RED
DROP
```

```
Management0 3.3.3.3 default
3.3.3.3 Vrf-RED
DROP
```

The index key must contain an interface or next-hop-ip value, or both values, or DROP for a gNMI or REST API request to access a route. The key value is not stored in OpenConfig database.

## View IP routes

You can view the IPv4 and IPv6 routes in the IP routing table using show commands.

### View IPv4 routes

#### Syntax

```
show ip route [vrf vrf-name] {static | connected | bgp |
ospf | summary [ipv4-prefix/mask]}
```

#### Parameters

- **vrf vrf-name** — (Optional) Enter a VRF instance name to filter the display of IPv4 routes in the IP routing table.
- **static** — Display user-configured static routes.
- **connected** — Display routes to a directly connected neighbor.
- **bgp** — Display BGP routes.
- **ospf** — Display OSPF routes.
- **summary** — Display summary information about entries in the IP routing table.

```
sonic# show ip route
Codes: K - kernel route, C - connected, S - static, B - BGP, O - OSPF
> - selected route, * - FIB route, q - queued route,
r - rejected route, # - not installed in hardware
Destination Gateway Dist/Metric Uptime

C>* 10.0.0.0/8 Direct Management0 0/0 08:41:42
S# 181.180.0.0/16 Direct Eth1/21 1/0 02:06:33
B>* 21.0.0.0/8 via 101.2.2.2 Vlan1012 20/0 01w1d10h
 via 102.3.3.2 Vlan1023
B>* 51.0.0.0/24 via 101.2.2.2 Vlan1012 20/0 01w1d10h
 via 102.3.3.2 Vlan1023
B>* 52.0.0.0/24 via 101.2.2.2 Vlan1012 20/0 01w1d10h
 via 102.3.3.2 Vlan1023
...
sonic# show ip route summary
```

| Route     | Source | Routes | FIB (vrf default) |
|-----------|--------|--------|-------------------|
| connected |        | 3      | 3                 |
| static    |        | 1      | 1                 |
| ebgp      |        | 8      | 8                 |
| ibgp      |        | 0      | 0                 |
| Totals    |        | 12     | 12                |

```
sonic# show ip route vrf Vrf_blue
Codes: K - kernel route, C - connected, S - static, B - BGP, O - OSPF
> - selected route, * - FIB route, q - queued route,
r - rejected route, # - not installed in hardware
Destination Gateway Dist/Metric Uptime

B>* 70.1.1.0/24 via 110.7.7.4 Vlan1307 20/0 01w1d08h
C>* 110.7.7.0/24 Direct Vlan1307 0/0 01w1d08h
```

### View IPv6 routes

```
show ipv6 route [vrf vrf-name] {static | connected | bgp | summary [ipv6-prefix/mask]}
```

#### Parameters

- **vrf vrf-name** — (Optional) Enter a VRF instance name to filter the display of IPv6 routes in the IP routing table.
- **static** — Display user-configured static routes.
- **connected** — Display IPv6 routes to a directly connected neighbor.
- **bgp** — Display BGP IPv6 routes.
- **summary** — Display summary information about entries in the IPv6 routing table.

```
sonic# show ipv6 route
Codes: K - kernel route, C - connected, S - static, B - BGP, O - OSPF
 > - selected route, * - FIB route, q - queued route,
 r - rejected route, # - not installed in hardware
 Destination Gateway Dist/Metric Uptime

C>* 1001:2222::/64 Direct Vlan1012 0/0 01w2d23h
C>* 1002:3333::/64 Direct Vlan1023 0/0 01w2d23h
S>* 1101:180::/64 Direct Eth1/22 1/0 19:41:32
B>* 5001:1111::/64 via fe80::92b1:1cff:fef4:ab9b Vlan1012 20/0 01w2d23h
 via fe80::92b1:1cff:fef4:ab9b Vlan1023
B>* 5002:2222::/64 via fe80::92b1:1cff:fef4:ab9b Vlan1012 20/0 01w2d23h
 via fe80::92b1:1cff:fef4:ab9b Vlan1023
C* fe80::/64 Direct Vlan1023 0/0 01w2d23h
C* fe80::/64 Direct Vlan1012 0/0 01w2d23h
C>* fe80::/64 Direct Management0 0/0 01w3d00h
...
...
```

```
sonic# show ipv6 route summary
Route Source Routes FIB (vrf default)
connected 5 5
static 1 1
ebgp 2 2
ibgp 0 0

Totals 8 8
```

```
sonic# show ipv6 route vrf Vrf_blue
Codes: K - kernel route, C - connected, S - static, B - BGP, O - OSPF
 > - selected route, * - FIB route, q - queued route,
 r - rejected route, # - not installed in hardware
 Destination Gateway Dist/Metric Uptime

S>* 1110:666::/64 Direct Vlan1307 1/0 14:18:17
C>* 1110:7777::/64 Direct Vlan1307 0/0 14:37:44
K>* 1110:7777::/128 Direct Vlan1307 0/0 14:37:44
C>* fe80::/64 Direct Vlan1307 0/0 14:37:44
K>* fe80::/128 Direct Vlan1307 0/0 14:37:44
K>* ff00::/8 Direct Vlan1307 0/256 14:37:44
```

## Policy-based routing

Policy-based routing (PBR) provides a method to forward packets by overriding the information available in the IP routing table. You can implement policies that selectively cause packets to take different paths.

Traditional IP routing forwards packets based only on the destination IP address in the packet. PBR can be configured to forward packets based on other criteria, such TCP/UDP port numbers, source IP address, DSCP value, and TCP flags. Enterprise SONiC uses flow-based service policies for policy-based routing.

Forwarding policies consist of class maps that select packets and set actions that cause a packet to be forwarded to a predetermined next hop or interface, bypassing the path determined by routing and forwarding tables. You can define multiple match and egress interface and/or next-hop values in the same policy. You can apply forwarding policies to switched or routed traffic. Traffic can be routed to the same VRF used by the ingress interface or a different VRF.

Use policy-based routing to provide equal access, protocol-sensitive routing, source-sensitive routing, routing based on interactive compare with batch traffic, and routing based on dedicated links. Policy-based routing is a more flexible mechanism for routing packets than destination routing.

## PBR forwarding policies

Enterprise SONiC implements PBR by providing a modular framework to classify traffic and apply forwarding actions, such as set IP next-hop, on selected traffic. To configure PBR:

1. Classify (select) traffic for policy-based routing by using ACLs or the L2, L3, or L4 fields in packet headers.
2. In a policy map, configure the forwarding actions to take on each classified flow.
3. Apply the forwarding policy on ingress interfaces — globally on all switch interfaces, a specified interface, a VLAN, or a port channel.

## Classify PBR traffic

To select traffic for policy-based routing, use an ACL or L2-L4 header classifiers.

### Classify traffic using modular ACLs

To classify traffic using modular ACLs:

1. Create an L2, IPv4, or IPv6 ACL to identify a traffic flow.

```
sonic(config)# {mac | ip | ipv6} access-list name
```

2. Add permit and deny rules to the ACL for L2 MAC, IPv4, or IPv6 traffic — see [Configure ACLs](#); for example:

```
Create IP ACL
sonic(config)# ip access-list pbr_v4_acl
sonic(conf-ipv4-acl)# seq 1 permit ip any 89.0.0.0/24 remark RULE_1
sonic(conf-ipv4-acl)# seq 2 permit ip any 89.0.1.0/24 remark RULE_2
sonic(conf-ipv4-acl)# seq 3 permit ip any 89.0.2.0/24 remark RULE_3
sonic(conf-ipv4-acl)# seq 4 permit ip any 89.0.3.0/24 remark RULE_4
sonic(conf-ipv4-acl)# seq 5 permit ip any 89.0.4.0/24 remark RULE_5
```

3. Create a classifier (class map) of match-type acl.

```
sonic(config)# class-map name match-type acl
```

Add the L2, IPv4, or IPv6 ACL to the class map to select flow traffic. Each class map uses only one ACL: L2 MAC, IPv4, or IPv6; for example:

```
Create class map for PBR IPv4 traffic
sonic(config)# class-map pbr_v4_class match-type acl
sonic(conf-class-map)# match access-group ip pbr_v4_acl
```

**(i) NOTE:** For a class map to be considered active, the ACL must be already configured. If the ACL is not configured, the classifier is incomplete and inactive. The class-map configuration is saved, and no error is displayed. When you configure the ACL, the classifier becomes active and applies any actions configured a policy.

### Classify traffic using L2-L4 header fields

For more fine-grained traffic classification in a flow, use match statements on L2, L3, and L4 header field values. You can combine match criteria for fields in different headers. For example, you can specify source MAC Address, VLAN, destination IP address, and TCP flags to identify a flow for forwarding actions. ACLs do not support this level of detailed packet classification. A class map is considered invalid if you configure mutually exclusive header fields, such as an IPv4 and an IPv6 address, as match criteria. If you enter no L2-L4 match statements in a class map, the classifier matches any traffic by default.

To classify traffic using L2-L4 header fields:

1. Create a classifier (class map) of match-type fields match-all.

```
sonic(config)# class-map name match-type fields match-all
```

- Add match statements to select packets based on L2, L3, and L4 header values — see [Configure flow-based ACLs](#); for example:

```
sonic(config)# class-map pbr_classmap match-type fields match-all
sonic(config-class-map)# match vlan 1001
sonic(config-class-map)# match destination-address mac host 00:01:00:11:00:11
sonic(config-class-map)# match destination-address ip 1.1.1.0/24
sonic(config-class-map)# match ip protocol tcp
sonic(config-class-map)# match tcp-flags syn rst
```

## Configure and apply PBR forwarding policies

A PBR forwarding policy specifies the forwarding actions to take on matching traffic for policy-based routing. A forwarding policy supports the following actions:

- Set next hop — Routes IPv4 traffic to an IPv4 next-hop; routes IPv6 traffic to an IPv6 next-hop.
- Set next-hop group — Specifies the group from which the best next-hop IPv4 or IPv6 address is chosen.
- Set interface — Forwards L2 traffic to a specified egress interface.
- Set interface null — Drops matching traffic if the null interface is set or if none of the specified next-hops are reachable or if the specified egress interface is not L2 and link up.

### PBR forwarding policies — Configuration notes

In Enterprise SONiC:

- A forwarding policy is supported only on ingress interfaces.
- Forwarding policies can only forward selected traffic; they cannot trap, switch, or route traffic to the CPU.
- Forwarding policies with next-hop and next-hop-group actions apply only on routed L3 traffic.
- Forwarding policies which use an Ethernet or port-channel egress interface apply only on switched L2 traffic.
- Forwarding policies do not apply on traffic destined to the CPU (destination IP address is the same as the switch address) or traffic which is trapped to the CPU.

### Configure PBR forwarding policy

- Create classifiers to select traffic using an L2, IPv4, or IPv6 ACL or L2-L4 packet header fields — see [Classify PBR traffic](#).
- Create a PBR policy map to configure the forwarding actions to take on classified traffic. The policy-map name must begin with an alphanumeric character; 63 characters maximum. It can contain alphanumeric, hyphen (-), and underscore (\_) characters.

```
sonic(config)# policy-map name type forwarding
```

- In policy-map flow configuration mode, add a class map to the policy. Enter a priority number (0-4095) to specify the order in which a class map is applied in the policy map to match traffic in the flow. A higher priority class map is processed before a lower priority.

```
sonic(config-policy-map)# class class-map-name priority number
sonic(config-policy-map-flow) #
```

- In policy-map flow configuration mode, add any of the following forwarding actions to take on classified traffic:

- Set the next-hop IPv4 address of matching traffic.

```
sonic(conf-policy-map-flow) # set ip next-hop ip-address [vrf vrf-name] [priority number]
```

To remove the configured IPv4 next-hop, enter the `no set ip next-hop ip-address` command.

- `next-hop ip-address` — Enter the next-hop IPv4 address. The next hop can be reached through the underlay network or a VXLAN tunnel.
- (Optional) `vrf vrf-name` — Enter the VRF used for the next-hop address. By default, the VRF of the interface on which the forwarding policy is applied is used or the default VRF if the next-hop is applied on all interfaces.
- (Optional) `priority number` — Enter the priority of the next-hop (1-65535; default is the lowest priority 0). The next-hop or next-hop group with the highest priority is used to forward traffic. If both a next-hop address and group have the highest priority, the setting that was configured first is used.

IPv4 next-hops are valid only if the class map uses a MAC ACL, IPv4 ACL, or IPv4 header fields to match traffic. MAC header fields can be used as additional matching criteria to IPv4 header fields. Only IPv4 routed traffic is forwarded to the configured next-hop. Combining IPv4 next-hops with IPv6 next-hops or an egress interface setting (except `set`

`interface null`) is not supported. To be selected for routing, the next-hop must be reachable. If none of the IPv4 next-hops are reachable, you can configure the null egress interface setting as a default drop action. If you do not configure the null egress interface, traffic is routed normally. The next-hop IP address should not be a local interface.

- Set the next-hop IPv6 address of matching traffic.

```
sonic(conf-policy-map-flow) # set ipv6 next-hop ipv6-address [vrf vrf-name] [priority number]
```

To remove the configured IPv6 next-hop, enter the `no set ipv6 next-hop ipv6-address` command.

- `next-hop ipv6-address` — Enter the next-hop IPv6 address. The next hop can be reached through the underlay network or a VXLAN tunnel.
- (Optional) `vrf vrf-name` — Enter the VRF used for the next-hop address. By default, the VRF of the interface on which the forwarding policy is applied is used or the default VRF if the next-hop is applied on all interfaces.
- (Optional) `priority number` — Enter the priority of the next-hop (1-65535; default is the lowest priority 0). The next-hop or next-hop group with the highest priority is used to forward traffic. If both a next-hop address and group have the highest priority, the setting that was configured first is used.

IPv6 next-hops are valid only if the class map uses a MAC ACL, IPv6 ACL, or IPv6 header fields to match traffic. MAC header fields can be used as additional matching criteria to IPv6 header fields. Only IPv6 routed traffic is forwarded to the configured next-hop. Combining IPv6 next-hops with IPv4 next-hops or an egress interface setting (except `set interface null`) is not supported. To be selected for routing, the next-hop must be reachable. If none of the IPv6 next-hops are reachable, you can configure the null egress interface setting as a default drop action. If you do not configure the null egress interface, traffic is routed normally. The next-hop IPv6 address should not be a local interface.

- Set the egress interface for matching traffic.

```
sonic(conf-policy-map-flow) # set interface {Eth slot/port[/breakout-port] | PortChannel number} [priority net number]
```

To remove the configured egress interface, enter the `no set interface {Eth slot/port[/breakout-port] | PortChannel number}` command.

- `Eth slot/port[/breakout-port] | PortChannel number` — Enter the egress port or port-channel number.
- (Optional) `priority number` — Enter the priority of the egress interface (1-65535; default is the lowest priority 0). The next-hop or next-hop group with the highest priority is used to forward traffic. If both a next-hop address and group have the highest priority, the setting that was configured first is used.

Setting an egress interface is valid only if the class map uses an L2 MAC ACL, IPv4 ACL, IPv6 ACL, or matching header fields. Only L2 switched traffic is forwarded to the specified egress interface. Combining an egress interface with IPv4 or IPv6 next-hops or next-hop groups is not supported. To be selectable for forwarding, the egress interface must be a switchport and online; if not, the interface is forward referenced. Ensure that the egress interface is a member of the required VLANs. If none of the egress interfaces are online, you can configure the null egress interface as a default drop action. If you do not configure the null egress interface, traffic is forwarded normally. If none of the IPv6 next-hops are reachable, you can configure the null egress interface setting as a default drop action. If you do not configure the null egress interface, traffic will be routed normally.

- Set the null egress interface as the default drop action in the forwarding policy. When configured, the drop action is the lowest priority action in the forwarding policy. The drop action is applied only if none of the configured next-hops or egress interfaces can be used for forwarding.

```
sonic(conf-policy-map-flow) # set interface null
```

To remove the configured null interface, enter the `no set interface null` command.

- Set an IPv4 next-hop group to use on matching traffic — see [Configure next-hop groups](#).

```
sonic(conf-policy-map-flow) # set ip next-hop-group name [priority number]
```

To remove the configured IPv4 next-hop group, enter the `no set ip next-hop-group name` command.

- `next-hop-group name` — Enter the name of an IPv4 next-hop group.
- (Optional) `priority number` — Enter the priority of the next-hop group (1-65535; default is the lowest priority 0). The next-hop or next-hop group with the highest priority is used to forward traffic. If both a next-hop address and group have the highest priority, the setting that was configured first is used.

An IPv4 next-hop group is valid only if the class map uses an L2 MAC ACL, or header fields to match traffic. Only IPv4 routed traffic is forwarded to an address in the configured next-hop group. Combining IPv4 next-hops with IPv6 next-hops or an egress interface setting (except `set interface null`) is not supported. To be selected for routing, the next-hop group must be online. If none of the IPv4 next-hops in the group are reachable, you can configure the

null egress interface setting as a default drop action. If you do not configure the null egress interface, traffic is routed normally.

- Set an IPv6 next-hop group to use on matching traffic — see [Configure next-hop groups](#).

```
sonic(conf-policy-map-flow) # set ipv6 next-hop-group name [priority number]
```

To remove the configured IPv6 next-hop group, enter the `no set ipv6 next-hop-group name` command.

- `next-hop-group name` — Enter the name of an IPv6 next-hop group.
- (Optional) `priority number` — Enter the priority of the next-hop group (1-65535; default is the lowest priority 0). The next-hop or next-hop group with the highest priority is used to forward traffic. If both a next-hop address and group have the highest priority, the setting that was configured first is used.

An IPv6 next-hop group is valid only if the class map uses an L2 MAC ACL, IPv6 ACL, or header fields to match traffic. Only IPv6 routed traffic is forwarded to an address in the configured next-hop group. Combining IPv6 next-hops with IPv64next-hops or an egress interface setting (except `set interface null`) is not supported. To be selected for routing, the next-hop group must be online. If none of the IPv4 next-hops in the group are reachable, you can configure the null egress interface setting as a default drop action. If you do not configure the null egress interface, traffic is routed normally.

5. Repeat Steps 3 and 4 to match traffic in a flow and apply forwarding actions on the selected traffic — see [Forwarding policy advanced configuration](#).
6. Apply a PBR forwarding policy map globally on all switch interfaces, a specified interface, a VLAN, or a port channel. To remove a policy from an interface, enter the `no` version of the command. You can apply forwarding policies only on ingress interfaces.
  - Globally on all switch interfaces:

```
sonic(config)# service-policy type forwarding in policy-map-name
```

**i | NOTE:** When you apply a forwarding policy globally on all interfaces, the next-hops must be in the default VRF unless you set a non-default VRF using the `set {ip | ipv6} next-hop ip-address vrf vrf-name` command in the policy map.

- On an interface or subinterface:

```
sonic(config)# interface Eth slot/port[/breakout-port] [.subinterface]
sonic(config-if-Eth)# service-policy type forwarding in policy-map-name
sonic(config-subintf-Eth)# service-policy type forwarding in policy-map-name
```

- On VLAN interfaces:

```
sonic(config)# interface Vlan vlan-id
sonic(conf-if-Vlan)# service-policy type forwarding in policy-map-name
```

- On port-channel interfaces:

```
sonic(config)# interface PortChannel portchannel-number
sonic(conf-if-po)# service-policy type forwarding in policy-map-name
```

## Forwarding policy advanced configuration

In a PBR forwarding policy, you can configure unique next-hops and backup next-hops.

### Configure unique next-hops

A policy can have multiple sections. Each section consists of a class map and associated actions in `set` statements. Also, each class map has a priority that indicates the order in which the classified traffic and actions are applied in the forwarding policy. You can set unique next-hops for different traffic flows by configuring multiple class-map classifiers with different priorities; for example:

```
sonic(config)# policy-map pbr_policy01 type forwarding
sonic(config-policy-map)# class pbr_class01 priority 100
sonic(config-policy-map-flow) # set ip next-hop 5.1.1.1
sonic(config-policy-map-flow) # exit
sonic(config-policy-map)# class pbr_class02 priority 90
sonic(config-policy-map-flow) # set ip next-hop 5.1.1.2
```

```
sonic(config-policy-map-flow) # exit
sonic(config-policy-map) #
```

## Configure backup next-hops

In a class map, you can configure backup next-hops by specifying different priorities for each `set ip next-hop` statement. If the active higher priority next-hop goes down, the next backup next-hop is used for forwarding; for example:

```
sonic(config) # policy-map pbr_policy type forwarding
sonic(config-policy-map) # class pbr_class
sonic(config-policy-map-flow) # set ip next-hop 5.1.1.1 priority 100
sonic(config-policy-map-flow) # set ip next-hop 5.1.1.2 priority 90
sonic(config-policy-map-flow) # set ip next-hop 5.1.1.3 priority 80
```

## Configure next-hop groups

A policy-based forwarding (PBF) next-hop group bundles a set of next-hops for load sharing and is used in a policy map assigned to interfaces. Unlike ECMP, a next-hop group can consist of paths which have different speeds. Use next-hop groups to optimize packet forwarding on interfaces. A next-hop group determines the best egress interface according to the reachability of an IPv4/IPv6 address over an underlay or VXLAN overlay network.

### Next-hop groups — Configuration notes

- A next-hop group consists of either IPv4 or IPv6 next-hops. Next-hop group members must be all IPv4 or IPv6. IPv6 member next-hops in a IPv4 next-hop group, or IPv4 next-hops in an IPv6 group, are not supported.
- Next-hop group members are used as egress for forwarding traffic. Each member is identified with a unique entry ID. You can configure duplicate next-hops with different entry IDs.
- For each next-hop member, you can specify an optional VRF for inter-VRF routing. If no VRF is configured, the VRF of the interface on which the policy is applied is used.
- For each next-hop member, you can optionally specify if the member is considered part of the group only if the next-hop is directly or indirectly connected. By default, both directly and indirectly connected next-hops are supported.
- To set more control on traffic forwarding to a next-hop group, you can configure optional threshold values. A threshold value limits the group members that are eligible for forwarding. The next-hop must be reachable and match the direct/indirectly connected criteria, if any. Configure a threshold either as a percentage of group members that are eligible for forwarding or as an absolute number of eligible members.
  - Up threshold specifies the value above which the next-hop group is considered as online and forwardable; default 1.
  - Down threshold specifies the value below which the next-hop group is considered as offline and non-forwardable; default 0.
- Next-hop group members must be non-ECMP enabled. If a next-hop group member is reachable through multiple paths, it is not considered as part of the next-hop group and traffic is not load-shared through it.

### Next-hop group configuration

1. Create a policy-based forwarding (PBF) next-hop group. The type entry is mandatory the first time you create the group in order to enter PBF configuration mode. To delete a PBF group, enter the `no pbf next-hop-group name` command.

```
sonic(config) # pbf next-hop-group name [type {ip | ipv6}]
```

2. Add a next-hop address. Re-enter the command to add multiple next-hop addresses.

```
sonic(config-pbf-next-hop-group) # entry entry-id next-hop ip-address [vrf vrf-name]
[recursive | non-recursive | overlay]
```

- `entry-id` — An ID number from 1 to 65535.
- `ip-address` — IPv4 or IPv6 address of the next hop reachable over an underlay or VXLAN overlay network; maximum 128 next-hop addresses.
- `vrf vrf-name` — VRF used to reach the next hop address. By default, the VRF to which an interface belongs is used. If the PBR policy map is configured globally on the switch, the default VRF is used.
- `recursive` — The next-hop IP address is in a route in the routing table and may not be directly connected to the switch.
- `non-recursive` — The next-hop IP address is not directly connected.
- `overlay` — The next-hop IP address is reachable through a VXLAN tunnel.

3. (Optional) Configure next-hop group thresholds. You can specify the number or percentage of next-hops above which the next-hop group is online for forwarding, or below which the group is offline.

```
sonic(config-pbf-next-hop-group)# threshold [type {count | percentage}] [up up-threshold] [down down-threshold]
```

- *up-threshold* — For count, a number from 1 to 128; for percentage, the minimum percentage from 1 to 100; default 1.
- *down-threshold* — For count, a number from 0 to 127; for percentage, a number from 0 to 99; default 0.

4. Apply the next-hop group to a forwarding policy map. Enter the classifier for the selected traffic. Enter a next-hop group with an optional priority (1-65535; default is the lowest priority 0). You can enter multiple next-hop groups with different priorities. In a policy map, the next-hop or next-hop group with the highest priority is used to forward traffic — see [Configure and apply forwarding policies](#). If both a next-hop address and group have the highest priority, the setting that was configured first is used.

```
sonic(config)# policy-map name type forwarding
sonic(config-policy-map)# class-map name priority number
sonic(config-policy-map-flow)# set ip next-hop-group name [priority number]
```

### **Example: Next-hop group configuration**

```
Create PBF next-hop-group
sonic(config)# pbf next-hop-group pbr_ecmp_group01 type ip
sonic(config-pbf-ip-nh-group)# entry 1 next-hop 133.3.1.2
sonic(config-pbf-ip-nh-group)# entry 2 next-hop 133.3.2.2
sonic(config-pbf-ip-nh-group)# entry 3 next-hop 133.3.3.2
sonic(config-pbf-ip-nh-group)# exit

Configure next-hop-group in forwarding policy
sonic(config)# policy-map pbr_policy type forwarding
sonic(config-policy-map)# class pbr class priority 100
sonic(config-policy-map-flow)# set ip next-hop-group pbr_ecmp_group01
```

### **View next-hop group configuration and policy-map binding**

```
sonic# show pbf next-hop-group [name | type {ip | ipv6}]

sonic# show pbf next-hop-group ipv4-test
Next-hop-group ipv4-test Type ip
 Description:
 Threshold type: percentage
 Threshold up: 80
 Threshold down: 30
 Members:
 entry 1 next-hop 10.1.1.1 recursive
 entry 2 next-hop 10.1.1.2 vrf VrfRed non-recursive
 entry 3 next-hop 10.1.1.3
 Referenced in flows:
 policy-map pbr-test at priority 100
```

### **View next-hop group status on an interface**

```
sonic# show pbf next-hop-group status {Eth slot/port[/breakout-port][.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch} [name | type {ip | ipv6}]

sonic# show pbf next-hop-group status Eth1/1
Eth 1/1
 Next-hop-group ipv4-test Type ip
 Status: Active
 Members:
 Entry 1 next-hop 10.1.1.1 recursive (Active)
 Entry 2 next-hop 10.1.1.2 vrf VrfRed non-recursive
 Entry 3 next-hop 10.1.1.3 (Active)
```

## Example: PBR forwarding configuration

```
Create class map using IPv4 ACL
sonic(config)# class-map pbr_class match-type acl
sonic(config-class-map)# match access-group ip pbr_acl
sonic(config-class-map)# exit

Create class map using match-type fields option
sonic(config)# class-map pbr_class02 match-type fields match-all
sonic(config-class-map)# match destination-address ip 2.1.1.0/24
sonic(config-class-map)# exit

Create class map with backup next-hops
sonic(config)# class-map pbr_class03
sonic(config-class-map)# set ip next-hop 5.1.1.1 priority 100
sonic(config-class-map)# set ip next-hop 5.1.1.2 priority 90
sonic(config-class-map)# set ip next-hop 5.1.1.3 priority 80
sonic(config-class-map)# exit

Create forwarding policy using ACL-based and match-type fields class maps
sonic(config)# policy-map pbr_policy type forwarding
sonic(config-policy-map)# class pbr_class priority 100
sonic(config-policy-map-flow)# set ip next-hop 5.1.1.4
sonic(config-policy-map-flow)# exit
sonic(config-policy-map)# class pbr_class02 priority 200
sonic(config-policy-map)# class pbr_class03 priority 300

View policy configuration
sonic# show running-configuration policy-map pbr_policy
!
policy-map pbr_policy type forwarding
class pbr_class priority 100
 set ip next-hop 5.1.1.1
class pbr_class priority 200
 match destination-address ip 2.1.1.0/24
class pbr_class priority 300
 set ip next-hop 5.1.1.1 priority 100
 set ip next-hop 5.1.1.1 priority 90
 set ip next-hop 5.1.1.1 priority 80

Apply forwarding policy on interface
sonic# interface Eth 1/3
sonic(conf-if-Eth1/3)# service-policy type forwarding in pbr_policy

Apply forwarding policy globally on all interfaces
sonic(config)# service-policy type forwarding in pbr_policy

Create class map using match-type fields option
sonic(config)# class-map pbr_class02 match-type fields match-all
sonic(config-class-map)# match destination-address ip 2.1.1.0/24
sonic(config-class-map)# end
sonic# show running-configuration class-map pbr_class02
!
class-map pbr_class02 match-type fields match-all
 match destination-address ip 2.1.1.0/24
```

## View PBR forwarding configuration

### View forwarding policy configuration

```
sonic# show service-policy policy-map {name | type forwarding}

sonic# show service-policy policy-map pbr_v4_policy
Eth1/1
 Policy pbr_v4_policy type forwarding at ingress
 Description:
 Flow pbr_v4_class at priority 100 (Active)
```

```
Description:
set ip nexthop 155.100.0.1 (Selected)
Packet matches: 0 frames 0 bytes
```

```
sonic# show service-policy policy-map pbr_v5_policy
!
policy-map pbr_v5_policy type forwarding
class pbr_v4_class priority 100
 set ip next-hop 155.100.0.3 priority 65535
 set ip next-hop 155.100.0.1 priority 1
```

### View forwarding policy binding

```
sonic# show service-policy summary [interface {Eth slot/port[/breakout-port]
[.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch} type
forwarding
```

```
sonic# show service-policy summary
Vlan1001
 forwarding policy pbr_v4_policy at ingress
Vlan1002
 forwarding policy pbr_v4_policy at ingress
Vlan1003
 forwarding policy pbr_v4_policy at ingress
```

### View forwarding policy binding and counters

```
sonic# show service-policy {interface {Eth slot/port[/breakout-port][.subinterface] |
PortChannel number[.subinterface] | Vlan vlan-id | Switch} type forwarding | policy-map
name [Eth slot/port[/breakout-port][.subinterface] | PortChannel number[.subinterface]
| Vlan vlan-id | Switch]
```

```
SONiC# show service-policy interface Vlan 100 type forwarding
Vlan100
Policy pbr_policy_example Type forwarding at ingress
Description:
Flow acl_class_1000 at priority 1000 (Active)
 Description:
 set ip next-hop 10.1.1.1 vrf default
 set ip next-hop 20.1.1.1 vrf VrfRed
 set ip next-hop 30.1.1.1 (Selected)
 set interface null
 Packet matches: 128 frames 128000 bytes
Flow acl_class_999 at priority 999 (Active)
 Description:
 set ip next-hop 11.1.1.1 vrf default (Selected)
 set ip next-hop 21.1.1.1 vrf VrfRed
 set ip next-hop 31.1.1.1
 set interface null
 Packet matches: 0 frames 0 bytes
Flow fields_class_0 at priority 999 (Active)
 Description:
 set ip next-hop 1111::1 vrf default
 set ip next-hop 2222::1 vrf VrfRed (Selected)
 set ip next-hop 3333::1
 set interface null
 Packet matches: 0 frames 0 bytes
```

To clear policy counters, enter the command:

```
sonic# clear counters service-policy {interface {Eth slot/port[/breakout-port]
[.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch}
type forwarding | policy-map name [Eth slot/port[/breakout-port][.subinterface] |
PortChannel number[.subinterface] | Vlan vlan-id | Switch]
```

## PBR quick configuration

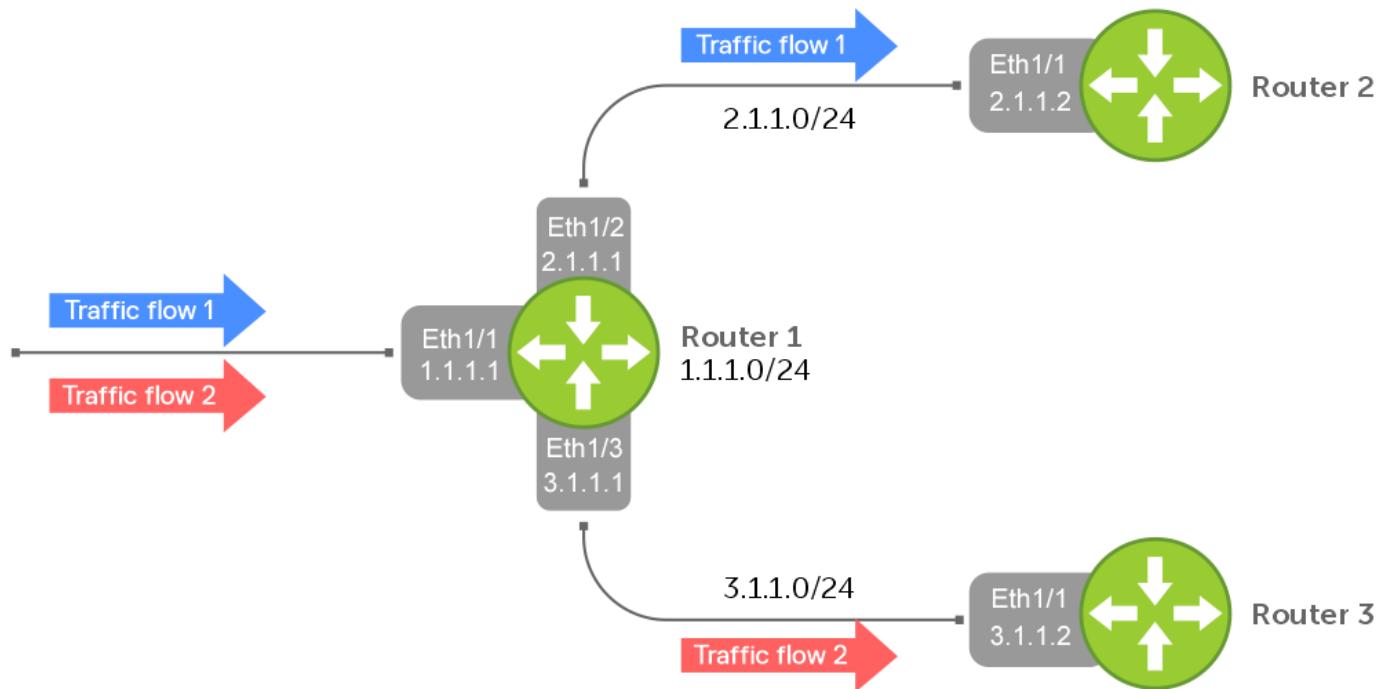


Figure 6. PBR topology

The following example shows how to configure PBR on Router 1. The Eth 1/1 interface on Router 1 receives two traffic flows with source IP addresses 1.1.1.2 and 1.1.1.3. The PBR configuration matches these traffic flows and forwards them to different next-hop Routers 2 and 3.

```
Configure PBR on Router 1
sonic(config)# class-map flow01 match-type fields match-all
sonic(config-class-map)# description match_flow_1
sonic(config-class-map)# match source-address ip host 1.1.1.2
sonic(config-class-map)# exit

sonic(config)# class-map flow02 match-type fields match-all
sonic(config-class-map)# description match_flow_2
sonic(config-class-map)# match source-address ip host 1.1.1.3
sonic(config-class-map)# exit

sonic(config)# policy-map pbr_rule type forwarding
sonic(config-policy-map)# class flow01 priority 100
sonic(config-policy-map-flow)# description send_to_R2
sonic(config-policy-map-flow)# set ip next-hop 2.1.1.2
sonic(config-policy-map-flow)# exit
sonic(config-policy-map)# class flow02 priority 90
sonic(config-policy-map-flow)# description send_to_R3
sonic(config-policy-map-flow)# set ip next-hop 3.1.1.2
sonic(config-policy-map-flow)# exit
sonic(config-policy-map)# exit

sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# service-policy type forwarding in pbr_rule
sonic(conf-if-Eth1/1)# do show service-policy policy-map pbr_rule
Eth1/1
 Policy pbr_rule type forwarding at ingress
 Description:
 Flow flow01 at priority 100 (Active)
 Description: send_to_R2
 set ip nexthop 2.1.1.2
 Packet matches: 100 frames 10000 bytes
 Flow flow02 at priority 90 (Active)
 Description: send_to_R3
```

```
set ip nexthop 3.1.1.2
Packet matches: 200 frames 20000 bytes
```

## Virtual Router Redundancy Protocol

Virtual Router Redundancy Protocol (VRRP) allows you to form virtual routers from groups of physical routers on your local area network (LAN). These virtual routing platforms — master and backup pairs — provide redundancy during hardware failure. VRRP also allows you to easily configure a virtual router as the default gateway to all your hosts. It also avoids the single point of failure of a physical router.

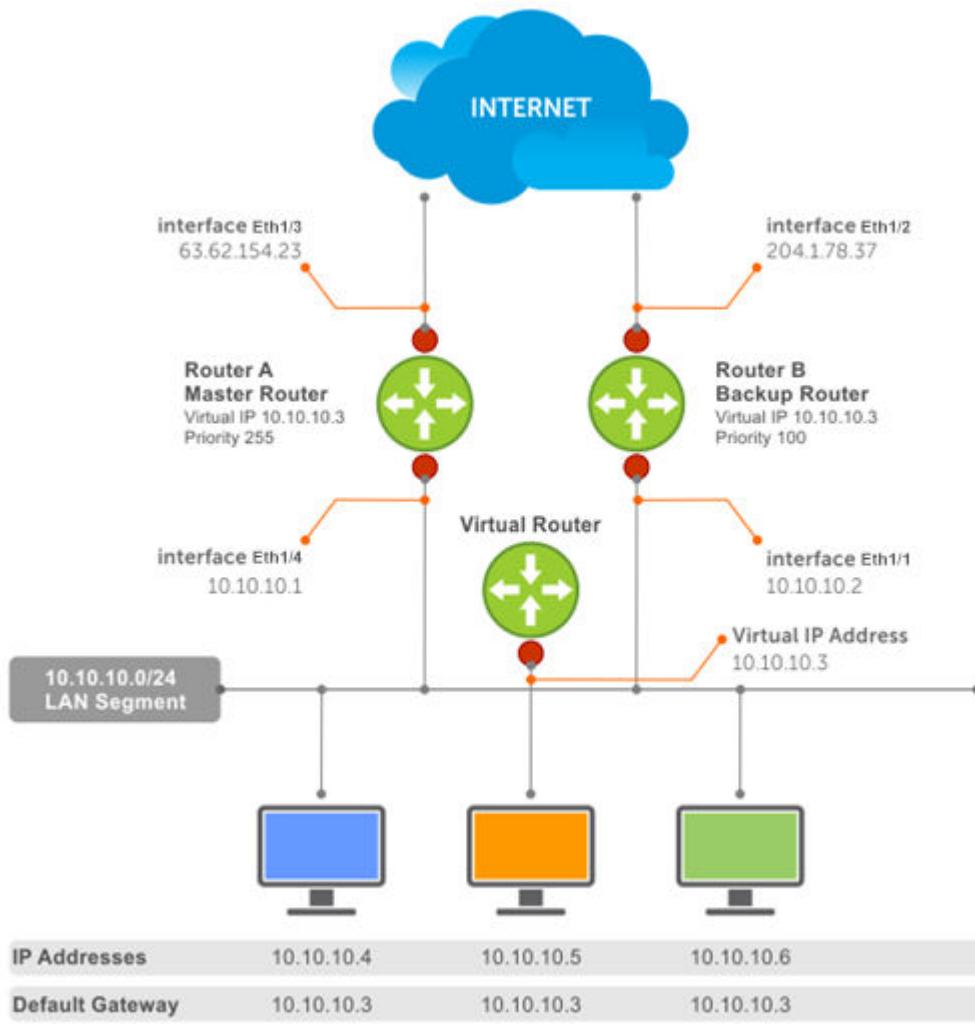
- Provides a virtual default routing platform
- Provides load balancing
- Supports multiple logical IP subnets on a single LAN segment
- Enables simple traffic routing without the single point of failure of a static default route
- Avoids issues with dynamic routing and discovery protocols
- Takes over a failed default router:
  - Within a few seconds
  - With minimum to no traffic loss
  - Without any interaction from hosts

## VRRP configuration

VRRP specifies a master, or active, router that owns the next-hop IP and MAC address for end stations on a LAN. The master router is chosen from the virtual routers by an election process and forwards packets sent to the next-hop IP address. If the master router fails, VRRP begins the election process to choose a new master router which continues routing traffic.

VRRP packets transmit with the virtual router MAC address as the source MAC address. The virtual router MAC address associated with a virtual router is in 00:00:5E:00:01:{VRID} format for IPv4 and 00:00:5E:00:02:{VRID} format for IPv6. The VRID is the virtual router identifier that allows up to 255 IPv4 and IPv6 VRRP routers on a network. The first four octets are unquenchable, the last two octets are 01:{VRID} for IPv4 and 02:{VRID} for IPv6. The final octet changes depending on the VRRP virtual router identifier.

### Basic VRRP Configuration



The example shows a typical network configuration using VRRP. Instead of configuring the hosts on network 10.10.10.0 with the IP address of either Router A or Router B as the default router, the default router of all hosts is set to the IP address of the virtual router. When any host on the LAN segment requests Internet access, it sends packets to the IP address of the virtual router.

Router A is configured as the master router with the virtual router IP address and sends any packets addressed to the virtual router to the Internet. Router B is the backup router and is also configured with the virtual router IP address.

If Router A, the master router, becomes unavailable (the connection between the LAN segment and Router A on Eth 1/1/6 goes down), Router B, the backup router, automatically becomes the master router and responds to packets sent to the virtual IP address. All workstations continue to use the IP address of the virtual router to transmit packets destined to the Internet. Router B receives and forwards packets on interface Eth 1/1/5. Until Router A resumes operation, VRRP allows Router B to provide uninterrupted service to the users on the LAN segment accessing the Internet.

When the interface that Router A uses to provide gateway services (Eth 1/1/7) goes down, Router B does not take over automatically. For Router B to become the master router, you must configure interface tracking. When you configure tracking on the interface and the interface goes down, the VRRP group's priority decreases. The lowered priority of the VRRP group triggers an election and Router B becomes the master router. See [Interface object tracking](#) for more information.

## Create virtual router

VRRP uses the VRID to identify each virtual router configured. Before using VRRP, you must configure the interface with the primary IP address.

- Create a virtual router for the interface with the VRRP identifier (1 to 255), then enter the address-family interface name.

```
sonic(conf-if-Eth1/2)# vrrp vrrp-id address-family afi-name
```

- Delete a VRRP identifier and address-family.

```
sonic(conf-if-Eth1/2) # no vrrp vrrp-id address-family afi-name
```

## Verify VRRP

```
sonic# show vrrp
Interface_Name VRID State VIP Cfg_Prio
Curr_Prio
Eth1/2 1 Master 40.0.0.5 120 120
```

```
sonic# show vrrp6
Interface_Name VRID State VIP Cfg_Prio
Curr_Prio
Eth1/2 1 Master 40::5 120 120
Eth1/3 2 Backup 80::5 100 100
```

## Delete VRRP

```
sonic(conf-if-Eth1/2) # no vrrp 1 address-family ipv4
```

```
sonic(conf-if-Eth1/2) # no vrrp 1 address-family ipv6
```

## Group version

Configure a VRRP version for the system. Define either VRRPv2 — `version 2` (default) or VRRPv3 — `version 3`.

- Configure a VRRP version for IPv4.

```
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # version 3
```

- Delete a VRRP version.

```
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # no version
```

## Virtual IP addresses

Virtual routers contain virtual IP addresses configured for that VRRP group (VRID). A VRRP group does not transmit VRRP packets until you assign the virtual IP address to the VRRP group.

To activate a VRRP group on an interface, configure at least one virtual IP address for a VRRP group. The virtual IP address is the IP address of the virtual router and does not require an IP address mask. You can configure up to 10 virtual IP addresses on a single VRRP group (VRID).

These rules apply to virtual IP addresses:

- The virtual IP addresses must be in the same subnet as the primary or secondary IP addresses configured on the interface. Though a single VRRP group can contain virtual IP addresses belonging to multiple IP subnets configured on the interface, Dell Technologies recommends configuring virtual IP addresses belonging to the same IP subnet for any one VRRP group. An interface on which you enable VRRP contains a primary IP address of 50.1.1.1/24 and a secondary IP address of 60.1.1.1/24. The VRRP group (VRID 1) must contain virtual addresses belonging to subnet 50.1.1.0/24 or subnet 60.1.1.0/24.
- If you configure multiple VRRP groups on an interface, only one of the VRRP groups can contain the interface primary or secondary IP address.

## Configure virtual IP address

Configure the virtual IP address — the primary IP address and the virtual IP addresses must be on the same subnet.

1. Configure a VRRP group (1 to 255).

```
sonic(conf-if-Eth1/2) # vrrp vrrp-id
```

- Configure virtual IP address for this VRRP ID (up to 10 IP addresses).

```
sonic(conf-if-Eth1/2) # vip vip-addr
```

#### View VRRP information

```
sonic# show vrrp
Interface_Name VRID State VIP Cfg_Prio
Curr_Prio
Eth1/2 1 Master 40.0.0.5 120 120
Eth1/3 2 Backup 80.0.0.5 100 100
```

#### View VRRP group 1

```
sonic# show vrrp interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 2
State is Master
Virtual IP address:
40.0.0.5
Virtual MAC address is 0000.5e00.0101
Track interface:
None
Configured Priority is 100, Current Priority is 100
Advertisement interval is 1 sec
Preemption is enabled
```

## Configure virtual IP addresses in a VRF

You can configure a VRRP group in a non-default VRF instance, and assign a virtual address to this group.

- Create the non-default VRF in which you want to configure VRRP.

```
sonic(config) # ip vrf vrf-name
```

- Enter the specific interface information. The interface must be in L3 mode.

```
sonic(conf-vrf-vrf1) # interface phy-if-name number
```

- Assign the interface to the non-default VRF that you have created.

```
sonic(conf-vrf-vrf1) # ip vrf forwarding vrf-name
```

- Assign an IP address to the interface.

```
sonic(conf-vrf-vrf1) # ip address ip-address
```

- Configure a VRRP identifier and address-family interface name.

```
sonic(conf-vrf-vrf1) # vrrp vrrp-id address-family afi-name
```

- Configure a virtual IP address for the VRRP ID.

```
sonic(conf-vrf-vrf1) # vip vip-addr
```

#### Configuration

```
sonic(config) # ip vrf Vrf1
sonic(conf-vrf-vrf1) # interface Eth1/2
sonic(conf-if-Eth1/2) # ip vrf forwarding Vrf1
sonic(conf-if-Eth1/2) # ip address 10.1.1.1/24
sonic(conf-if-Eth1/2) # vrrp 1 address-family ipv4
sonic(conf-if-Eth1/2) # vip 40.0.0.5
```

#### Verify configuration

```
sonic# show vrrp
Interface_Name VRID State VIP Cfg_Prio
```

```
Curr_Prio
Eth1/2 1 Master 40.0.0.5 120 120
Eth1/3 2 Backup 80.0.0.5 100 100
```

## Set group priority

The router that has the highest primary IP address of the interface becomes the master. The default priority for a virtual router is 100. If the master router fails, VRRP begins the election process to choose a new master router based on the next-highest priority. The virtual router priority is automatically set to 255, if any of the configured virtual IP addresses matches the interface IP address.

1. Create a virtual router for the interface with the VRRP identifier (1 to 255).

```
sonic(conf-if-Eth1/2)# vrrp vrrp-id
```

2. Configure the priority number for the VRRP group (1 to 254; default 100).

```
sonic(conf-if-Eth1/2-vrrp-ipv4-1)# priority number
```

### Configuration

```
sonic(conf-if-Eth1/2)# vrrp 1 address-family ipv4
sonic(conf-if-Eth1/2-vrrp-ipv4-1)# priority 120
```

```
sonic(conf-if-Eth1/2)# vrrp 1 address-family ipv6
sonic(conf-if-Eth1/2-vrrp-ipv6-1)# priority 120
```

### Verify configuration

```
sonic# show vrrp
Interface_Name VRID State VIP Cfg_Prio
Curr_Prio
Eth1/2 1 Master 40.0.0.5 120 120
Eth1/3 2 Backup 80.0.0.5 100 100
```

```
sonic# show vrrp6
Interface_Name VRID State VIP Cfg_Prio
Curr_Prio
Eth1/2 1 Master 40::5 120 120
Eth1/3 2 Backup 80::5 100 100
```

### Verify VRRP group priority

```
sonic# show vrrp interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 2
State is Master
Virtual IP address:
40.0.0.5
Virtual MAC address is 0000.5e00.0101
Track interface:
None
Configured Priority is 100, Current Priority is 100
Advertisement interval is 1 sec
Preemption is enabled
```

```
sonic# show vrrp6 interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 3
State is Master
Virtual IP address:
40::5
Virtual MAC address is 0000.5e00.0201
Track interface:
None
Configured Priority is 100, Current Priority is 100
```

```
Advertisement interval is 1 sec
Preemption is enabled
```

## Disable preempt

Prevent the backup router with the higher priority from becoming master router by disabling the preemption process. The `preempt` command is enabled by default and forces the system to change the master router if another router with a higher priority comes online.

You must configure all virtual routers in the VRRP group with the same settings. Configure all routers with `preempt` enabled or configure all with `preempt` disabled.

1. Configure a virtual router for the interface with the VRRP identifier (1 to 255).

```
sonic(conf-if-Eth1/2) # vrrp vrrp-id
```

2. Prevent any backup router with a higher priority from becoming the Master router.

```
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # no preempt
```

### Configuration

```
sonic(conf-if-Eth1/2) # vrrp 1 address-family ipv4
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # no preempt
```

```
sonic(conf-if-Eth1/2) # vrrp 1 address-family ipv6
sonic(conf-if-Eth1/2-vrrp-ipv6-1) # no preempt
```

### Verify configuration

```
sonic# show vrrp interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 2
State is Master
Virtual IP address:
40.0.0.5
Virtual MAC address is 0000.5e00.0101
Track interface:
None
Configured Priority is 100, Current Priority is 100
Advertisement interval is 1 sec
Preemption is disabled
```

```
sonic# show vrrp6 interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 3
State is Master
Virtual IP address:
40::5
Virtual MAC address is 0000.5e00.0201
Track interface:
None
Configured Priority is 100, Current Priority is 100
Advertisement interval is 1 sec
Preemption is disabled
```

## Advertisement interval

By default, the master router transmits a VRRP advertisement to all members of the VRRP group every one second, indicating it is operational and is the master router.

If the VRRP group misses three consecutive advertisements, the election process begins and the backup virtual router with the highest priority transitions to master. To avoid throttling VRRP advertisement packets, Dell Technologies recommends increasing the VRRP advertisement interval to a value higher than the default value of one second. If you change the time interval between VRRP advertisements on one router, change it on all participating routers.

- Create a virtual router for the interface with the VRRP identifier (1 to 255).

```
sonic(conf-if-Eth1/2) # vrrp vrrp-id
```

- Change the advertisement interval setting in seconds (1 to 255; default 1).

```
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # advertise-interval seconds
```

## Configuration

```
sonic(conf-if-Eth1/2) # vrrp 1 address-family ipv4
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # advertise-interval 1
```

```
sonic(conf-if-Eth1/2) # vrrp 1 address-family ipv6
sonic(conf-if-Eth1/2-vrrp-ipv6-1) # advertise-interval 1
```

## Verify advertisement interval

```
sonic# show vrrp interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 2
State is Master
Virtual IP address:
40.0.0.5
Virtual MAC address is 0000.5e00.0101
Track interface:
None
Configured Priority is 100, Current Priority is 100
Advertisement interval is 1 sec
Preemption is enabled
```

```
sonic# show vrrp6 interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 3
State is Master
Virtual IP address:
40::5
Virtual MAC address is 0000.5e00.0201
Track interface:
None
Configured Priority is 100, Current Priority is 100
Advertisement interval is 1 sec
Preemption is enabled
```

## Interface tracking

You can monitor the state of any interface according to the virtual group. If the tracked interface goes down, the VRRP group priority decreases by a default value of 10 — also known as cost. If the tracked interface's state goes up, the VRRP group priority increases by the priority cost.

The lowered priority of the VRRP group may trigger an election. As the master/backup VRRP routers are selected based on the VRRP group's priority, tracking features ensure that the best VRRP router is the active for that group. The combined priority of the VRRP router is all its tracking interface must be less than 254. If you configure the VRRP group as the owner router with a priority 255, tracking for that group is disabled, regardless of the state of the tracked interfaces. The priority of the owner group always remains 255.

For a virtual group, track the line-protocol state of any interface using the `interface` command. Enter an interface type and slot/port[/breakout-port][:subport] information, or VLAN number.

- Ethernet — Physical interface
- Vlan — VLAN interface, from 1 to 4093
- Loopback — Loopback interface, from 0 to 16383

For a virtual group, track the status of a configured object using the `track` command and the object number. You can also configure a tracked object for a VRRP group with this command before you create the tracked object. No changes in the VRRP group priority occur until the tracked object is determined to be down.

## Configure interface tracking

1. Create a virtual router for the interface with the VRRP identifier (1 to 255).

```
sonic(conf-if-Eth1/2) # vrrp vrrp-id
```

2. Increase the effective priority by weight value if the track interface is up.

```
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # track-interface interface-name {weight wt_value}
```

### Configuration

```
sonic(conf-if-Eth1/2) # vrrp 1 address-family ipv4
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # track-interface Eth1/4 weight 10
```

```
sonic(conf-if-Eth1/2) # vrrp 1 address-family ipv6
sonic(conf-if-Eth1/2-vrrp-ipv6-1) # track-interface Eth1/7 weight 10
```

### Disable tracking

```
sonic(conf-if-Eth1/2-vrrp-ipv4-1) # no track-interface Eth1/4
```

```
sonic(conf-if-Eth1/2-vrrp-ipv6-1) # no track-interface Eth1/7
```

### Verify VRRP group priority

```
sonic# show vrrp interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 2
State is Master
Virtual IP address:
40.0.0.5
Virtual MAC address is 0000.5e00.0101
Track interface:
None
Configured Priority is 100, Current Priority is 100
Advertisement interval is 1 sec
Preemption is enabled
```

```
sonic# show vrrp6 interface Eth1/2 vrid 1
Eth1/2, VRID 1
Version is 3
State is Master
Virtual IP address:
40::5
Virtual MAC address is 0000.5e00.0201
Track interface:
None
Configured Priority is 100, Current Priority is 100
Advertisement interval is 1 sec
Preemption is enabled
```

## Network Address Translation

**i | NOTE:** Network Address Translation (NAT) is available only in the Cloud Standard, Cloud Premium, Enterprise Standard, and Enterprise Premium bundles. NAT is not available in the Edge Standard bundle.

Network Address Translation enables the process that assigns a public IP address to devices that access resources outside the network. NAT conserves IP address usage in the local network.

NAT is not required within the network to route traffic between private IP addresses. A NAT gateway translates the private IP addresses of local network devices to a globally unique, public IP address when they communicate with remote devices.

**i | NOTE:** Network Address Translation is not supported on the Z9432F-ON.

## Enable NAT

To enable NAT and enter NAT configuration commands in NAT Configuration mode:

```
sonic(config)# nat
sonic(conf-nat)# enable
```

To disable NAT, use no enable.

## NAT configuration

After you enable NAT, you can perform these steps in any order:

- Add a static NAT entry.
- Add a static NAT entry with an L4 port.
- Create a NAT address pool.
- Configure NAT binding.
- Configure NAT zones.
- Configure dynamic NAT timeout.

### Add static NAT entry

To communicate outside your network over the Internet, you can manually configure a static NAT entry to replace a local IP address with a globally unique IP address.

```
sonic(conf-nat)# static basic global-ip local-ip [snat | dnat] [twice_nat_id value]
```

- Source NAT (snat) — Translates a source IP address in the local network to a global IP address sent to an external network. Remote devices in outside networks use the global address to access the local device.
- Destination NAT (dnat) — Translates a destination IP address in packets, which are received from an external network and traverse the local network, into a local IP address used in the local network. dnat is the default.
- twice\_nat\_id value — Performs address translation on both source and destination IP addresses for static entries which have the same value .
- To remove a static NAT entry, use no static basic global-ip local-ip.

```
sonic(conf-nat)# static basic 125.4.4.4 12.1.1.1
sonic(conf-nat)# static basic 100.100.100.100 15.15.15.15 snat twice-nat-id 5
sonic(conf-nat)# static basic 200.200.200.5 17.17.17.17 dnat twice-nat-id 5
```

### Add static NAT entry with L4 port

You can also configure a static NAT entry to translate a local IP address and TCP or UDP port number into a global IP addresses with TCP or UDP port number.

```
sonic(conf-nat)# static {tcp | udp} global-ip global-port local-ip local-port
[snat | dnat] [twice_nat_id value]
```

- Source NAT (snat) — Translates a source IP address and TCP/UDP port in the local network into a global IP address and TCP/UDP port that is sent to an external network. Remote devices in outside networks use the global address and L4 port to access the local device.
- Destination NAT (dnat) — Translates a destination IP address and TCP/UDP port in packets, which are received from an external network and traverse the local network, into a local IP address and TCP/UDP port used in the local network. dnat is the default.
- twice\_nat\_id value — Performs address translation on both source and destination IP addresses for static entries which have the same ID value .

```
sonic(conf-nat)# static udp 148.56.7.7 8991 10.11.1.12 2000
sonic(conf-nat)# static tcp 123.3.4.1 901 11.11.1.1 1000
sonic(conf-nat)# static tcp 65.55.46.6 106 20.0.0.6 206 dnat twice-nat-id 200
sonic(conf-nat)# static tcp 65.55.45.5 100 20.0.0.5 200 snat twice-nat-id 200
```

### Create NAT address pool

If you configure dynamic NAT, a local address is replaced using a pool of global addresses. Dynamic translation is useful when multiple users on a private network access the Internet. Configure a pool of available global addresses by defining the global IP address range, and optionally the TCP/UDP port range used for local address translation.

```
pool pool-name global-ip-range [global-port-range]
```

- Enter the global IP address range in the format *ip-address-ip-address*.
- Enter the TCP/UDP port-number range in the format *portnumber-port-number*.
- To delete a NAT address pool, use the no pool *pool-name* command. To delete all NAT address pools, use no pools.

```
sonic(conf-nat)# pool Pool1 19.19.19.19
sonic(conf-nat)# pool Pool2 20.0.0.7 1024-65535
sonic(conf-nat)# pool Pool3 65.55.45.10-65.55.45.15 500-1000
```

## Configure NAT binding

```
binding binding-name pool-name [acl-name] [snat | dnat] [twice_nat_id value]
```

When you configure an address pool binding:

- snat — Translates a source IP address to a global IP address in the pool. snat is the default setting in NAT binding.
- dnat — Translates a destination IP address to a global IP address in the pool.
- twice\_nat\_id value — Performs address translation on both source, and destination IP addresses using the address pool for static entries which have the same ID *value*.

To limit the IP addresses in a global NAT address pool, you can use an access control list (ACL). By default, if you specify an ACL, traffic from all IP hosts is allowed. A permit statement allows the IP addresses that have the attributes configured in the permit rule. A deny statement denies packets that have the attributes configured in the deny rule. In an ACL, the do\_not\_nat entry allows packets to be routed instead of translated.

```
sonic(conf-nat)# binding Bind1 Pool1 10_ACL_IPV4
sonic(conf-nat)# binding Bind2 Pool2 12_ACL_IPV4 snat twice-nat-id 25
sonic(conf-nat)# binding Bind3 Pool3 15_ACL_IPV4 dnat twice-nat-id 25
```

To remove the ACL-pool binding, enter no binding *binding-name*. To remove all ACL-pool bindings, enter no bindings.

## Configure NAT zones

Configure a NAT zone on L3 interfaces so that NAT address translation is performed on packets when a packet traverses a zone on configured interfaces. You can configure a NAT zone on any Ethernet, VLAN, port channel, or loopback interface that is configured with an IP address. The range of NAT zone numbers is from 0 to 3.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# ip address 20.20.20.20/24
sonic(conf-if-Eth1/2)# nat-zone 1
sonic(conf-if-Eth1/2)# exit
sonic(config)# interface Vlan 5
sonic(conf-if-Vlan5)# ip address 23.23.23.23/24
sonic(conf-if-Vlan5)# nat-zone 1
sonic(conf-if-Vlan5)# exit
sonic(config)# interface PortChannel 2
sonic(conf-if-po2)# ip address 25.25.25.25/24
sonic(conf-if-po2)# nat-zone 1
sonic(conf-if-po2)# exit
sonic(config)# interface Loopback 1
sonic(conf-if-lol1)# ip address 10.10.10.10/32
sonic(conf-if-lol1)# nat-zone 2
```

The source zone of a packet is the zone of the inbound interface on which the packet is received. The destination zone of the packet is the zone of the L3 next-hop interface from the L3 route lookup of the destination address.

- Inbound traffic entering on a source interface is L3 forwarded using static DNAT translation.
- Outbound traffic being sent on a destination interface is dynamically SNAT translated.

To remove the NAT configuration on all interfaces, enter no nat interfaces.

## Configure dynamic NAT timeout

By default, only dynamic address translation configurations time out after 10 minutes (600 seconds) of having no active traffic. There is no timeout for static NAT entries. To change the dynamic NAT timeout value, specify a new value in seconds (300 to 432000).

```
sonic(config)# nat
sonic(conf-nat)# timeout 1200
```

You can also change the NAT entry timeouts for L4 UDP and TCP NAT entries. To change the TCP timeout for address translation, enter a new timeout value in seconds (300 to 432000; default 86400). To change the UDP timeout for address translation, enter a new value in seconds (120 to 600; default 300).

```
sonic(config)# nat
sonic(conf-nat)# udp-timeout 600
sonic(conf-nat)# tcp-timeout 66460
```

## View NAT configuration

Use these commands to display the NAT configuration and operation and NAT table entries.

### View NAT translation table

```
sonic# show nat translations
```

| Protocol | Source        | Destination     | Translated Source | Translated Destination |
|----------|---------------|-----------------|-------------------|------------------------|
| all      | 10.0.0.1      | ---             | 65.55.42.2        | ---                    |
| all      | ---           | 65.55.42.2      | ---               | 10.0.0.1               |
| all      | 10.0.0.2      | ---             | 65.55.42.3        | ---                    |
| all      | ---           | 65.55.42.3      | ---               | 10.0.0.2               |
| tcp      | 20.0.0.1:4500 | ---             | 65.55.42.1:2000   | ---                    |
| tcp      | ---           | 65.55.42.1:2000 | ---               | 20.0.0.1:4500          |
| udp      | 20.0.0.1:4000 | ---             | 65.55.42.1:1030   | ---                    |
| udp      | ---           | 65.55.42.1:1030 | ---               | 20.0.0.1:4000          |
| tcp      | 20.0.0.1:6000 | ---             | 65.55.42.1:1024   | ---                    |
| tcp      | ---           | 65.55.42.1:1024 | ---               | 20.0.0.1:6000          |
| tcp      | 20.0.0.1:5000 | 65.55.42.1:2000 | 65.55.42.1:1025   | 20.0.0.1:4500          |
| tcp      | 20.0.0.1:4500 | 65.55.42.1:1025 | 65.55.42.1:2000   | 20.0.0.1:5000          |

To clear the entries in the NAT translation table, use `clear nat translations`.

### View NAT translation statistics

```
sonic# show nat statistics
```

| Protocol | Source              | Destination       | Packets | Bytes |
|----------|---------------------|-------------------|---------|-------|
| all      | 100.100.100.100     | 200.200.200.5     | 15      | 12785 |
| all      | 17.17.17.17         | 15.15.15.15       | 10      | 12754 |
| all      | 12.12.12.14         | ---               | 0       | 0     |
| all      | ---                 | 138.76.28.1       | 12      | 12500 |
| tcp      | 12.12.15.15:1200    | ---               | 0       | 0     |
| tcp      | ---                 | 138.76.29.2:250   | 8       | 85120 |
| tcp      | 100.100.101.101:251 | 200.200.201.6:276 | 21      | 21654 |
| tcp      | 17.17.18.18:1251    | 15.15.16.16:1201  | 18      | 21765 |

To clear the NAT statistics, use `clear nat statistics`.

### View NAT and NAPT configuration

```
sonic# show nat config static
```

| Nat  | Type | IP | Protocol | Global IP  | Global L4 Port | Local IP | Local L4 Port | Twice-Nat Id |
|------|------|----|----------|------------|----------------|----------|---------------|--------------|
| dnat | all  |    |          | 65.55.45.5 | ---            | 10.0.0.1 | ---           | ---          |
| dnat | all  |    |          | 65.55.45.6 | ---            | 10.0.0.2 | ---           | ---          |

```

dnat tcp 65.55.45.7 2000 20.0.0.1 4500 1
snat tcp 20.0.0.2 4000 65.55.45.8 1030 1

```

### **View NAT pools**

```

sonic# show nat config pool

Pool Name Global IP Range Global L4 Port Range
----- -----
Pool1 65.55.45.5 1024-65535
Pool2 65.55.45.6-65.55.45.8 ---
Pool3 65.55.45.10-65.55.45.15 500-1000

```

### **View NAT binding configuration**

```

sonic# show nat config bindings

Binding Name Pool Name Access-List Nat Type Twice-Nat Id
----- -----
Bind1 Pool1 --- snat ---
Bind2 Pool2 1 snat 1
Bind3 Pool3 2 snat --

```

### **View global NAT configuration**

```

sonic# show nat config globalvalues

Admin Mode : enabled
Global Timeout : 600 secs
TCP Timeout : 86400 secs
UDP Timeout : 300 secs

```

### **View L3 interface zones**

```

sonic# show nat config zones
Port Zone

Eth1/1 1
Loopback0 1
Vlan5 0
PortChannel12 2

```

### **View NAT translation counts**

```

sonic# show nat translations count

Static NAT Entries 4
Static NAPT Entries 2
Dynamic NAT Entries 0
Dynamic NAPT Entries 4
Static Twice NAT Entries 0
Static Twice NAPT Entries 4
Dynamic Twice NAT Entries 0
Dynamic Twice NAPT Entries 0
Total SNAT/SNAPT Entries 9
Total DNAT/DNAPPT Entries 9
Total Entries 14

```

## **ECMP**

Equal Cost Multi Path (ECMP) is a Layer 3 routing strategy to forward traffic to a destination using multiple available paths. ECMP increases the number of paths to a destination and increases the available bandwidth to reach the destination. The ECMP mechanism enables load balancing and increased bandwidth by using unused links and bandwidth.

### **ECMP hashing**

Although ECMP can increase the number of available links, all available paths should be closely load-shared so that no path is over or underutilized. You can modify the IP ECMP load-share hashing parameters to affect traffic load-sharing across multiple available paths.

To obtain optimal ECMP load sharing, configure ECMP in all devices that reside between the source and the destination.

### Configure ECMP for IPv4

- Use the following global command:

```
ip load-share hash ipv4 {ipv4-src-ip | ipv4-dst-ip | ipv4-ip-proto | ipv4-14-src-port
| ipv4-14-dst-port | symmetric}
```

### Configure ECMP for IPv6

- Use the following global command:

```
ip load-share hash ipv6 {ipv6-src-ip | ipv6-dst-ip | ipv6-next-hdr | ipv6-14-src-port
| ipv6-14-dst-port | symmetric}
```

### Configure ECMP hash seed

Configure a unique hash seed for each device to avoid hash polarization which may result in network congestion. Network polarization can happen when multiple data flows try to reach a switch using the same switch ports.

- Use the following global command:

```
ip load-share hash seed seed-value
```

The value for *seed-value* is from 0 to 16777215.

### View ECMP hashing mode

```
show ip load-share
IP Hash Mode: Default
IPv6 Hash Mode: Symmetric
Packet Header Fields:
IP: ipv4-src-ip ipv4-dst-ip ipv4-ip-proto ipv4-14-src-port ipv4-14-dst-port
IPv6: ipv6-src-ip ipv6-dst-ip ipv6-next-hdr ipv6-14-src-port ipv6-14-dst-port
Hash seed: 10
```

## IP helper

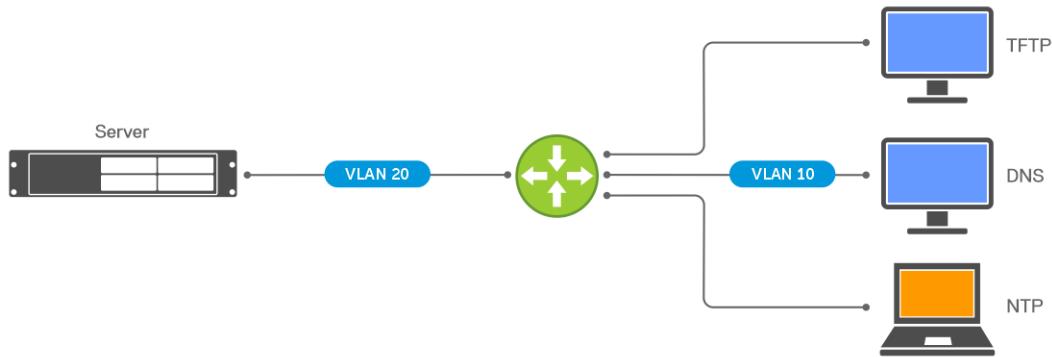
An IP helper helps devices in a network to identify a unicast server in a different network.

To communicate with other devices or avail certain network resources in a network, hosts or other devices generally use the unicast mechanism. In scenarios where the destination server is not known, devices send a broadcast request to identify the servers. This mechanism works in a broadcast domain.

When the server is present in a different broadcast domain then the clients, routers usually do not forward broadcast messages.

You can use the IP helper mechanism such that routers in your network behave as the relay and helps in forwarding the requests from the hosts to the configured servers.

In the following figure, hosts that require various services are connected to the device that is enabled with the IP helper mechanism. When the IP helper receives broadcast messages, if the UDP port is enabled for relaying, it forwards the message to the server.



Before configuring IP helper, enable UDP forwarding globally. When you configure the helper address on an interface and global UDP forwarding, by default, the UDP broadcasts that are sent on the following default ports are relayed:

- TFTP (port 69)
- DNS (port 53)
- NTP (37)
- NetBIOS Name Server (port 137)
- NetBIOS Datagram Server (port 138)
- TACACS service (port 49)

Along with the default ports, you can configure custom ports globally.

IP helper forwards packets if they meet the following criteria:

- Destination IP address is 255.255.255.255 or the subnet broadcast address of the receiving interface, such as, 10.10.10.255 for the interface configured with 10.10.10.10/24.
- Apart from the default ports, a UDP port is explicitly configured for forwarding.
- The IP TTL value is greater than or equal to 2.
- Helper addresses are configured on the incoming interface.
- UDP forwarding is enabled globally.

## Configure IP helper

To configure IP helper on the system, follow these steps:

- Enable UDP broadcast forwarding globally.

```
ip forward-protocol udp enable
```

- Add an IP helper address on an interface.

```
ip helper-address [vrf vrf-name] ip-address
```

- *vrf-name* - Enter the VRF name in which the server address is reachable.
- *ip-address* - Enter the IP address of the server.

- (Optional) Add a UDP protocol or port to the list of forwarding ports globally.

```
ip forward-protocol udp include {tftp | dns | ntp | netbios-name-server | netbios-datagram-server | tacacs | port}
```

- *tftp* - Enter to enable if it is disabled. This option is enabled by default.
- *dns* - Enter to enable if it is disabled. This option is enabled by default.
- *ntp* - Enter to enable if it is disabled. This option is enabled by default.
- *netbios-name-server* - Enter to enable if it is disabled. This option is enabled by default.
- *netbios-datagram-server* - Enter to enable if it is disabled. This option is enabled by default.
- *tacacs* - Enter to enable if it is disabled. This option is enabled by default.
- *port* - Enter a custom port number.

- (Optional) Remove a UDP protocol or port from the list of forwarding ports globally.

```
ip forward-protocol udp exclude {tftp | dns | ntp | netbios-name-server | netbios-
datagram-server | tacacs | port}
```

- tftp - Enter to disable. This option is enabled by default.
- dns - Enter to disable. This option is enabled by default.
- ntp - Enter to disable. This option is enabled by default.
- netbios-name-server - Enter to disable. This option is enabled by default.
- netbios-datagram-server - Enter to disable. This option is enabled by default.
- tacacs - Enter to disable. This option is enabled by default.
- port - Enter a custom port number.

- Configure the UDP broadcast packet rate limiting value globally.

```
ip forward-protocol udp rate-limit value-in-pps
```

- value-in-pps - Enter a PPS value. The range is from 600 to 10000. The default value is 600.

## View IP helper information

### Display the IP helper configuration

```
show ip helper-address [interface]
```

#### Example

```
show ip helper-address Eth1/7

Interface Vrf Relay address

Eth1/7 31.1.0.2
 2.2.2.3
vrf20 11.19.0.144
```

```
show ip helper-address

Interface Vrf Relay address

Eth1/7 31.1.0.2
 2.2.2.3
vrf20 11.19.0.144
Eth1/8 31.1.0.2
```

### View the IP helper global configuration

```
show ip forward-protocol
```

#### Example

```
show ip forward-protocol

UDP forwarding: Enabled
UDP rate limit: 6000 pps
UDP forwarding enabled on the ports: TFTP , NTP , 330 , 234, 1000
UDP forwarding disabled on the ports: DNS , NetBios-Name-server , NetBios-datagram-
server
```

### View the IP helper statistics

```
show ip helper-address statistics [interface]
```

## Example

```
show ip helper-address statistics Eth1/7

 Packets received : 1098
 Packets relayed : 980
 Packets dropped : 118
 Invalid TTL packets : 22
 All ones broadcast packets received : 602
 Net directed broadcast packets received : 496
```

```
show ip helper-address statistics

Eth1/7

 Packets received : 1098
 Packets relayed : 980
 Packets dropped : 118
 Invalid TTL packets : 22
 All ones broadcast packets received : 602
 Net directed broadcast packets received : 496

Eth1/8

 Packets received : 100
 Packets relayed : 90
 Packets dropped : 10
 Invalid TTL packets : 5
 All ones broadcast packets received : 50
 Net directed broadcast packets received : 50
```

## Clear relay statistics

```
clear ip helper-address statistics [interface]
```

# Multicast

IP multicast routing enables a host (source) to send packets to a group of hosts (receivers) anywhere within the IP network by using a special form of IP address called the IP multicast group address. The sending host inserts the multicast group address into the IP destination address field of the packet, and IP multicast routers and multilayer switches forward incoming IP multicast packets out all interfaces that lead to the members of the multicast group. Any host, regardless of whether it is a member of a group, can send to a group but only the members of a group can receive the message.

Multicast provides an efficient method for delivering traffic flows that can be characterized as one-to-many or many-to-many. In a multicast network, the key component is the routing device, which can replicate packets and is multicast-capable. The routing devices in the IP multicast network, which has the same topology as the unicast network it is based on, use a multicast routing protocol to build a distribution tree that connects receivers to sources.

These protocols support multicast in IPv4 networks for distribution:

- Internet group management protocol (IGMP)
- Protocol-independent multicast (PIM)

**i NOTE:**

- Enterprise SONiC supports IGMP and IPv4 PIM for multicast routing; this release of Enterprise SONiC does not support IPv6 PIM
- Enterprise SONiC supports PIM and IGMP on default and nondefault VRFs

**Topics:**

- Configure multicast routing
- Internet group management protocol
- Protocol-independent multicast

## Configure multicast routing

Configuring multicast routing is a two-step process that involves configuring multicast routing and enabling PIM sparse mode (PIM-SM) on a Layer 3 (L3) interface. For more information about IGMP and PIM feature configurations, see [Internet group management protocol](#) and [Protocol-independent multicast](#).

## Internet group management protocol

**i NOTE:** L3 IGMP is supported only in the Enterprise Standard, Enterprise Premium, and Edge Standard bundles. L3 IGMP is not supported in the Cloud Standard and Cloud Premium bundles.

The Internet group management protocol (IGMP) is a communications protocol that establishes multicast group memberships using IPv4 networks. Enterprise SONiC supports IGMPv1, IGMPv2, and IGMPv3 to manage the multicast group memberships on IPv4 networks.

A multicast router (mrouter) is a Layer 3 router or switch that has multicast features enabled. When a host requests to join a multicast group, it sends an IGMP message to the mrouting. Each network segment has an IGMP querier, which is an mrouting.

The IGMP querier periodically (by default, every 125 seconds) sends out a membership query to all the hosts. The hosts — in response to the query — send a response back to the querier to report their multicast group memberships. The switch makes an entry to identify the corresponding port as a member of the particular multicast group.

## Multicast routers

Multicast routers (mrouters) send these types of queries:

- General query — to learn about listeners for multicast groups

- Multicast address-specific query — to learn if a specific multicast address has listeners
- Multicast address-and-source-specific query — to learn if any of the sources from the specified list for a multicast source has any listeners

Hosts send these messages to mrouters:

- Version 1 — membership report
- Version 2
  - Version 1 membership report for backward-compatibility with Version 1
  - Version 2 membership report
  - Leave group message
- Version 3
  - Version 1 membership report for backward-compatibility with Version 1
  - Version 2 membership report for backward-compatibility with Version 2
  - Version 3 membership report
  - Version 2 leave-group message

Version 3 provides support for source filtering. The system reports interest in receiving packets only from specified source addresses, or from all sources except specified addresses, that are sent to a multicast address.

## Standards compliance

- Enterprise SONiC complies with RFCs 1112, 2236, and 3376 for IGMP versions 1, 2, and 3, respectively.
- Enterprise SONiC version 3 as the default IGMP version; version 3 is backwards compatible with versions 1 and 2.

## Important notes

- Enterprise SONiC cannot serve as an IGMP host or an IGMP version 1 querier.
- Enterprise SONiC automatically enables IGMP on interfaces where you enable PIM sparse mode.

## Supported IGMP versions

IGMP has three versions. Version 3 obsoletes and is backwards-compatible with version 2; version 2 obsoletes version 1. Enterprise SONiC supports these IGMP versions:

- Router — IGMP versions 2 and 3 (version 3 is the default)
- Host — IGMP versions 1, 2, and 3

In IGMP version 2, the host expresses interest in a particular group membership (\*, G). In IGMP version 3, the host expresses interest in a particular group membership, and specifies the source from which it wants the multicast traffic (S, G).

## Query interval

The IGMP querier periodically sends a general query to discover which multicast groups are active. A group must have at least one host to be active. By default, the periodic query messages are sent every 125 seconds.

```
sonic(config)# interface Vlan 120
sonic(conf-if-Vlan120)# ip igmp query-interval 60
```

## Last member query interval

When the IGMP querier receives a leave message, it sends a group-specific query message to ensure if any other host in the network are interested in the multicast flow. By default, the group-specific query messages are sent every 1000 milliseconds.

```
sonic(config)# interface Vlan 120
sonic(conf-if-Vlan120)# ip igmp last-member-query-interval 200
```

## Response timer

The maximum response time is the amount of time that the querier waits for a response to a query before taking action. When a host receives a query, it does not respond immediately, but rather starts a delay timer.

The delay time is set to a random value between 0 and the maximum response time. The host sends a response when the timer expires — in IGMP version 2, if another host responds before the timer expires, the timer nullifies, and no response is sent.

The querier advertises the maximum response time in the query. Lowering this value increases response burstiness because all host membership reports are sent before the maximum response time expires. Increasing this value decreases burstiness.

```
sonic(config)# interface Vlan 120
sonic(conf-if-Vlan120)# ip igmp query-max-resp-time 20
```

## Select IGMP version

Enterprise SONiC enables Version 3 by default. You can change the IGMP version to meet host requirements.

```
sonic(config)# interface Vlan 120
sonic(conf-if-Vlan120)# ip igmp version 2
```

## View IGMP-enabled interfaces and groups

To view IGMP-enabled interfaces and groups, use `show` commands.

### View IGMP-enabled interfaces

```
sonic# show ip igmp interface Vlan602

Interface : Vlan602
UpTime : 01:11:53
State : up
Address : 172.60.1.1
Version : 3
Querier

Querier : local
Start Count : 0
Query Timer : 00:01:33
Other Timer : -::--
Timers

Group Membership Interval : 260s
Last Member Query Count : 2
Last Member Query Time : 2s
Older Host Present Interval : 260s
Other Querier Present Interval : 255s
Query Interval : 125s
Query Response Interval : 10s
Robustness Variable : 2
Startup Query Interval : 31s
Flags

All Multicast : no
Broadcast : yes
Deleted : no
Interface Index : 484
Multicast : yes
Multicast Loop : 0
Promiscuous : no
```

## View IGMP-enabled groups

```
sonic# show ip igmp groups

Interface Address Group Mode Timer Srcs V Uptime
Vlan602 172.60.1.1 232.0.0.1 ---- 00:01:36 1 3 01:11:50
```

## IGMP snooping

**i | NOTE:** IGMP snooping is available only in the Enterprise Standard, Enterprise Premium, and Edge Standard bundles. IGMP snooping is not available in the Cloud Standard and Cloud Premium bundles.

IGMP snooping streamlines multicast traffic handling for VLANs. By examining (snooping) IGMP membership report messages from interested hosts, multicast traffic is limited to a subset of VLAN interfaces on which the hosts reside.

IGMP protocol messages are examined within a VLAN to discover which interfaces are connected to hosts or other devices that are interested in receiving this traffic. Using the interface information, IGMP snooping can reduce bandwidth consumption in a multiaccess LAN environment to avoid flooding the entire VLAN.

IGMP snooping tracks which ports are attached to multicast-capable routers to help it manage the forwarding of IGMP membership reports. This *snooping* then responds to topology change notifications.

**i | NOTE:** Regardless of IGMP snooping being enabled, unknown multicast data traffic is flooded all VLAN member ports.

## Configure IGMP snooping

### Enable IGMP snooping

Enables IGMP snooping on a per-VLAN basis. If the global setting is disabled, all VLANs are treated as disabled, whether they are enabled or not. Default is enabled.

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping
```

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping
```

### Enable IGMP querier

When there is no multicast router (mrouter) in the VLAN to originate queries, you must configure an IGMP snooping querier to send membership queries. When an IGMP snooping querier is enabled, it sends out periodic IGMP queries that trigger IGMP report messages from hosts that want to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping querier
```

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping querier
```

### Enable IGMP fast leave

Removes the group state when it receives an IGMP Leave report without sending an IGMP query message. This parameter is used for IGMPv2 hosts when no more than one host is present on each VLAN port. The default is disabled.

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping fast-leave
```

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping fast-leave
```

### Configure IGMP query interval

Defines the maximum response time advertised in IGMP group-specific queries. Set an appropriate value for the IGMP last member query interval to speed up host responses to IGMP group-specific queries and avoid IGMP report traffic bursts. A receiver host starts a report delay timer for a multicast group when it receives an IGMP group-specific query for the group. This timer is set to a random value in the range of 0 to the maximum response time advertised in the query. When the timer value decreases to 0, the host sends an IGMP report to the group. Range is 100 to 25500 milliseconds; default 1000.

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping query-interval 20
```

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping query-interval
```

### **Set IGMP last member query interval**

Sets the interval to wait after sending an IGMP query to verify that no hosts that want to receive a particular multicast group remain on a network segment. If no hosts respond before the last member query interval expires, the software removes the group from the associated VLAN port. Range is 100 to 25500 milliseconds; default 1000.

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping last-member-query-interval 2000
```

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping last-member-query-interval
```

### **Configure IGMP query max response time**

Configures a query maximum response time that is advertised on IGMP queries. Range is 1 to 25 seconds; default 10 (10000 milliseconds).

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping query-max-response-time 12
```

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping query-max-response-time
```

### **Configure IGMP version**

Configures the IGMP version. The device supports IGMPv1, IGMPv2, and IGMPv3. These versions are interoperable. For example, if IGMP snooping is enabled and the querier's version is IGMPv2, the device receives IGMPv3 report from a host, then the device can forward the IGMPv3 report to the mrouter. Range is 1 to 3; default 2.

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping version 3
```

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping version
```

### **Configure IGMP mrouter**

Configures a static connection to a multicast router (mrouter).

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping mrouter interface Eth1/2
```

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping mrouter interface Eth1/2
```

### **Configure IGMP multicast group**

Configures a Layer 2 port of a VLAN as a static member of an IGMP multicast group.

```
sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# ip igmp snooping static-group 225.0.0.1 interface PortChannel12

sonic(config)# interface Vlan120
sonic(conf-if-Vlan120)# no ip igmp snooping static-group 225.0.0.1 interface PortChannel12
```

### IGMP snooping configuration

```
sonic(config)# ip igmp snooping enable
sonic(config)# interface Vlan 120
sonic(conf-if-Vlan120)# ip igmp snooping mrouter interface Eth 1/1/1
sonic(conf-if-Vlan120)# ip igmp snooping querier
sonic(conf-if-Vlan120)# ip igmp version 3
sonic(conf-if-Vlan120)# ip igmp snooping fast-leave
sonic(conf-if-Vlan120)# ip igmp snooping query-interval 60
sonic(conf-if-Vlan120)# ip igmp snooping query-max-resp-time 10
sonic(conf-if-Vlan120)# ip igmp snooping last-member-query-interval 1000
```

### Verify IGMP snooping configuration

```
sonic# show ip igmp snooping
Vlan ID: 120
Querier: Disabled
IGMP Operation mode: IGMPv3
Is Fast-Leave Enabled: Disabled
Query interval: 125
Last Member Query Interval: 1000
Max Response time: 10

Vlan ID: 220
Querier: Enabled
IGMP Operation mode: IGMPv2
Is Fast-Leave Enabled: Disabled
Query interval: 125
Last Member Query Interval: 1000
Max Response time: 10

Vlan ID: 320
Querier: Enabled
IGMP Operation mode: IGMPv1
Is Fast-Leave Enabled: Disabled
Query interval: 20
Last Member Query Interval: 1000
Max Response time: 10
```

```
sonic# show ip igmp snooping vlan 120
Vlan ID: 120
Querier: Enabled
IGMP Operation mode: IGMPv3
Is Fast-Leave Enabled: Disabled
Query interval: 125
Last Member Query Interval: 1000
Max Response time: 10
```

## Protocol-independent multicast

**NOTE:** PIM source-specific multicast (SSM) is available only in the Enterprise Standard, Enterprise Premium, and Edge Standard bundles. PIM SSM is not available in the Cloud Standard and Cloud Premium bundles.

Protocol independent multicast (PIM) is a group of multicast routing protocols that provides one-to-many and many-to-many transmission of information. PIM uses routing information from other routing protocols and does not depend on any specific unicast routing protocol. PIM uses any unicast routing protocol that is deployed in the network. Enterprise SONiC supports the following modes:

PIM source-specific multicast—PIM SSM uses a subset of PIM sparse mode and IGMP version 3 (IGMPv3) to allow a client to receive multicast traffic directly from the source. PIM SSM uses the PIM sparse-mode functionality to create a shortest-path tree (SPT) between the receiver and the source. PIM SSM is simpler than PIM sparse mode because only the one-to-many model is supported. PIM SSM builds SPTs rooted at the source immediately because in SSM, the router closest to the interested receiver host is informed of the unicast IP address of the source for the multicast traffic.

## PIM terminology

|                                 |                                                                                                                           |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>(S, G)</b>                   | Source and Group address pair.                                                                                            |
| <b>Shortest path tree (SPT)</b> | Source-based trees where the multicast traffic routes are based on the shortest path between the source and the receiver. |
| <b>Outgoing interface (OIF)</b> | Interface through which a multicast packet is sent to the receiver.                                                       |
| <b>Incoming interface (IIF)</b> | Interface through which a multicast packet is received from the source.                                                   |

## PIM-SSM

PIM-SSM uses source-based trees. A separate multicast distribution tree is built for each multicast source that sends data to a multicast group. Each multicast distribution tree has a router near the source as its root node.

Sources send data directly to the root of the tree. PIM-SSM enables receivers to specify the source from which to receive data and the multicast group they want to join. The receiver identifies a multicast data stream using the source and group address pair (S, G).

**i NOTE:**

- PIM-SSM requires receivers to support IGMP Version 3.
- The default PIM-SSM range is 232.0.0.0/8; the default range is always supported and the range can never be smaller than the default.

## Configure SSM prefix-list

The standard PIM-SSM multicast range is 232.0.0.0/8. Apart from the standard range, you can configure the SSM prefix-list per VRF in CONFIGURATION mode. This value is used to qualify other multicast group addresses as the PIM-SIM range using an IP prefix-list. Use the `ip prefix-list` command to create the corresponding prefix-list, and then associate the prefix-list to the PIM.

**i NOTE:** This prefix-list cannot be deleted from the system until after removal of any PIM global configuration which refer to the prefix-list.

```
sonic(config)# ip pim ssm prefix-list pim_ssm_pfx_list
```

**i NOTE:** For PIM-SSM to work properly, the prefix-list must contain at least one valid entry. When removing the sequence number rule from the existing prefix-list, ensure not to remove the last sequence number rule if it is associated with the PIM-SSM range. Removing the last sequence number configuration from the prefix-list results in an empty prefix-list, which affects the corresponding association in the PIM-SSM range.

## PIM global configuration commands

These PIM configuration commands can be run per VRF. If the VRF is not specified, the commands are run as the default VRF.

Once PIM global configurations are done on a specific nondefault VRF, that VRF cannot be deleted from the device until the relevant PIM configurations are cleared. The `no` version of all commands removes the current configuration.

## Configure join-prune-interval

Configure the join-prune-interval in CONFIGURATION mode per VRF (60 to 600 seconds). VRF name is up to 15 characters.

```
sonic(conf) # ip pim join-prune-interval 70
```

## Configure keepalive timer

Configure the keepalive timer in seconds per VRF (31 to 60,000 seconds) in CONFIGURATION mode. This value is the period after the last (S, G) data packet during which the (S, G) join state will be even in the absence of the (S, G) join message.

```
sonic(conf) # ip pim keep-alive-timer 35
```

## Enable PIM ECMP

Enable PIM ECMP per VRF in CONFIGURATION mode. If the PIM has a choice of ECMP next-hops for a specific RPF, the PIM will cause the S, G flow to be spread out between the next-hops. If this command is not specified, the first next-hop found is used.

```
sonic(config) # ip pim ecmp
```

Enable PIM ECMP rebalance per VRF. If the PIM is using ECMP and an interface goes down, this command causes the PIM to rebalance all S, G flows across the remaining next-hops. If this command is not configured, the PIM only modifies those (S, G) flows that were using the interface that went down.

```
sonic(config) # ip pim ecmp rebalance
```

**i** **NOTE:** In order to build a forwarding tree, PIM employs the RPF procedure to select an upstream interface. If ECMP is enabled and the best path to the RPF neighbor changes or a path loss occurs over one of the ECMP paths, the `ip ecmp` and `ip ecmp-rebalance` commands should be configured so that PIM considers all available next hops and recalculates all stream paths.

## PIM interface-specific commands

All interface-specific PIM commands are supported on Ethernet, PortChannel, L3 subinterfaces, and VLAN interfaces. Once PIM interface-specific configurations are complete on an interface, that interface cannot be deleted until the PIM interface configurations exist on that interface. The interface can be deleted once the PIM interface configurations are cleared.

For dynamic port breakout, when a specific physical port is in break-in or breakout mode, the existing interfaces are deleted and a new set of interfaces are created. As part of that existing interface delete, all relevant PIM configurations on that specific interface are cleaned up internally.

VRF change for a specific interface is not allowed once the PIM interface-specific configurations are present in that interface. You must clean up before changing the VRF. The `no` version of these commands removes the current configuration.

## Enable PIM sparse-mode

Enable PIM on the required an interface.

```
sonic# configure terminal
sonic(conf) # interface Vlan100
sonic(conf-if-Vlan100) # ip pim sparse-mode
```

## Set the designated router priority

Set the designated router priority for the PIM interface (1 to 4294967295) in INTERFACE mode. This command sets the priority of a node to become a DR. A higher value means higher chances of being elected.

```
sonic(conf-if-Vlan100)# ip pim drpriority 10
```

## Set hello interval

Set the hello interval for hello messages to keep the PIM neighbor session alive in seconds (1 to 255; default is 30) in INTERFACE mode. This command configures the default hold-time (3.5 \* hello-interval), which is the period to keep the PIM neighbor session alive even without hello messages from that specific neighbor.

```
sonic(conf-if-Vlan100)# ip pim hello 30
```

## Enable BFD

Enable BFD support for PIM on the interface.

```
sonic(conf-if-Vlan100)# ip pim bfd
```

## Apply a BFD profile to an interface

Associate a BFD profile to an interface.

```
sonic(config)# bfd
sonic(conf-bfd)# profile test
sonic(conf-bfd)# exit
sonic(config)# interface Vlan100
sonic(conf-if-Vlan100)# ip pim bfd profile test
```

## Clear PIM configuration

### clear ip mroute

Resets all IP multicast routes for a specific VRF.

```
sonic(config)# clear ip mroute vrf Vrf1
```

### clear ip pim

Resets all PIM interfaces of a specific VRF.

```
sonic(config)# clear ip pim vrf Vrf1 interfaces
```

Rescans PIM outgoing interfaces list (OIL) of all multicast entries of a specific VRF.

```
sonic(config)# clear ip pim vrf Vrf1 oil
```

## View multicast PIM-SSM global configuration

View the authentication manager information for the interface.

Use these show commands to view the default VRF and the nondefault VRF. You can also view details about all VRFs using the vrf all option.

show ip mroute displays all IP multicast routes.

```
sonic# show ip mroute
IP Multicast Routing Table
 * -> indicates installed route

 Source Group Input Output Uptime
* 71.0.0.11 233.0.0.1 Vlan100 Vlan200 00:41:59
* 71.0.0.22 233.0.0.1 Vlan100 Vlan200 00:41:54
 Vlan201 00:41:59
* 71.0.0.11 234.0.0.1 Vlan100 Vlan200 00:41:34
* 71.0.0.33 234.0.0.1 Vlan100 Vlan200 00:41:31
 Vlan201 00:41:44
* 71.0.0.22 235.0.0.1 Vlan100 Vlan200 00:41:16
* 71.0.0.33 235.0.0.1 Vlan100 Vlan200 00:41:14
```

The following show command displays IP multicast routes for a group address (233.0.0.1).

```
sonic# show ip mroute 233.0.0.1
IP Multicast Routing Table
 * -> indicates installed route

 Source Group Input Output Uptime
* 71.0.0.11 233.0.0.1 Vlan100 Vlan200 00:41:59
* 71.0.0.22 233.0.0.1 Vlan100 Vlan200 00:41:54
 Vlan201 00:41:59
```

The following show command displays IP multicast routes for a group address (233.0.0.1) and source address (71.0.0.22).

```
nic# show ip mroute 233.0.0.1 71.0.0.22
IP Multicast Routing Table
 * -> indicates installed route

 Source Group Input Output Uptime
* 71.0.0.22 233.0.0.1 Vlan100 Vlan200 00:41:54
 Vlan201 00:41:59
```

The following show command displays IP multicast routes for a VRF.

```
sonic# show ip mroute vrf Vrf1
IP Multicast Routing Table for VRF: Vrf1
 * -> indicates installed route

 Source Group Input Output Uptime
* 51.0.0.11 233.0.0.1 Vlan300 Vlan301 00:41:59
* 51.0.0.22 233.0.0.1 Vlan300 Vlan301 00:41:54
 Vlan302 00:41:59
```

The following show command displays IP multicast routes for all VRFs.

```
ic# show ip mroute vrf all
IP Multicast Routing Table for VRF: default
 * -> indicates installed route

 Source Group Input Output Uptime
* 71.0.0.11 233.0.0.1 Vlan100 Vlan200 00:41:59
* 71.0.0.22 233.0.0.1 Vlan100 Vlan200 00:41:54
 Vlan201 00:41:59
* 71.0.0.11 234.0.0.1 Vlan100 Vlan200 00:41:34
* 71.0.0.33 234.0.0.1 Vlan100 Vlan200 00:41:31
 Vlan201 00:41:44
* 71.0.0.22 235.0.0.1 Vlan100 Vlan200 00:41:16
* 71.0.0.33 235.0.0.1 Vlan100 Vlan200 00:41:14
```

```

IP Multicast Routing Table for VRF: Vrf1
* -> indicates installed route

 Source Group Input Output Uptime
* 51.0.0.11 233.0.0.1 Vlan300 Vlan301 00:41:59
* 51.0.0.22 233.0.0.1 Vlan300 Vlan301 00:41:54
 Vlan302 00:41:59

```

The following show commands display IP multicast routes summary.

```

sonic# show ip mroute summary
IP Multicast Routing Table summary for VRF: default

Mroute Type Installed/Total
(S, G) 6/6

```

```

sonic# show ip mroute vrf Vrf1 summary
IP Multicast Routing Table summary for VRF: Vrf1

Mroute Type Installed/Total
(S, G) 2/2

```

```

sonic# show ip mroute vrf all summary
IP Multicast Routing Table summary for VRF: default

Mroute Type Installed/Total
(S, G) 6/6

IP Multicast Routing Table summary for VRF: Vrf1

Mroute Type Installed/Total
(S, G) 2/2

```

## show ip pim

```

show ip pim [vrf {vrf-name | all}] {[interface [ifName]]} | {[nbr [nbr-addr]]} | [rpf]
| [ssm] | {[topology {[grp-addr [src-addr]}]}]}

```

```

sonic# show ip pim interface
PIM interface information for VRF: default
Interface State Address PIM Nbrs PIM DR
Hello-interval PIM DR-Priority
Vlan100 up 100.0.0.2 1 100.0.0.2 30
 1
Vlan200 up 200.0.0.2 1 200.0.0.3 30
 1

```

```

sonic# show ip pim interface vlan 100
PIM interface information for VRF: default
Interface State Address PIM Nbrs PIM DR
Hello-interval PIM DR-Priority
Vlan100 up 100.0.0.2 1 100.0.0.2 30
 1

```

```

sonic# show ip pim vrf Vrf1 interface
PIM Interface information for VRF: Vrf1

Interface State Address PIM Nbrs PIM DR
Hello-interval PIM DR-Priority
Vlan300 up 30.0.0.2 1 30.0.0.2 30

```

```

sonic# show ip pim vrf all interface
PIM Interface information for VRF: default

```

| Interface                                 | State     | Address         | PIM Nbrs   | PIM DR       |
|-------------------------------------------|-----------|-----------------|------------|--------------|
| Hello-interval                            |           | PIM DR-Priority |            |              |
| Vlan100<br>1                              | up        | 100.0.0.2       | 1          | 100.0.0.2 30 |
| Vlan200<br>1                              | up        | 200.0.0.2       | 1          | 200.0.0.3 30 |
| PIM Interface information for VRF: Vrf1   |           |                 |            |              |
| Interface                                 | State     | Address         | PIM Nbrs   | PIM DR       |
| Hello-interval                            |           | PIM DR-Priority |            |              |
| Vlan300<br>1                              | up        | 30.0.0.2        | 1          | 30.0.0.2 30  |
| sonic# show ip pim neighbor               |           |                 |            |              |
| PIM Neighbor information for VRF: default |           |                 |            |              |
| Interface                                 | Neighbor  | Uptime          | Expirytime | DR-Priority  |
| Vlan100                                   | 100.0.0.1 | 01:38:52        | 00:01:22   | 1            |
| Vlan200                                   | 200.0.0.3 | 01:22:33        | 00:01:13   | 1            |
| sonic# show ip pim neighbor 100.0.0.1     |           |                 |            |              |
| PIM Neighbor information for VRF: default |           |                 |            |              |
| Interface                                 | Neighbor  | Uptime          | Expirytime | DR-Priority  |
| Vlan100                                   | 100.0.0.1 | 01:38:52        | 00:01:22   | 1            |
| sonic# show ip pim vrf Vrf1 neighbor      |           |                 |            |              |
| PIM Neighbor information for VRF: Vrf1    |           |                 |            |              |
| Interface                                 | Neighbor  | Uptime          | Expirytime | DR-Priority  |
| Vlan300                                   | 30.0.0.1  | 01:48:32        | 00:01:12   | 1            |
| sonic# show ip pim vrf all neighbor       |           |                 |            |              |
| PIM Neighbor information for VRF: default |           |                 |            |              |
| Interface                                 | Neighbor  | Uptime          | Expirytime | DR-Priority  |
| Vlan100                                   | 100.0.0.1 | 01:38:52        | 00:01:22   | 1            |
| Vlan200                                   | 200.0.0.3 | 01:22:33        | 00:01:13   | 1            |
| PIM Neighbor information for VRF: Vrf1    |           |                 |            |              |
| Interface                                 | Neighbor  | Uptime          | Expirytime | DR-Priority  |
| Vlan300                                   | 30.0.0.1  | 01:48:32        | 00:01:12   | 1            |
| sonic# show ip pim ssm                    |           |                 |            |              |
| PIM SSM information for VRF: default      |           |                 |            |              |
| SSM group range : PIM_PLIST1              |           |                 |            |              |
| sonic# show ip pim ssm                    |           |                 |            |              |
| PIM SSM information for VRF: default      |           |                 |            |              |
| SSM group range : 232.0.0.0/8             |           |                 |            |              |
| sonic# show ip pim vrf Vrf1 ssm           |           |                 |            |              |
| PIM SSM information for VRF: Vrf1         |           |                 |            |              |
| SSM group range : PIM_PLIST1              |           |                 |            |              |
| sonic# show ip pim vrf all ssm            |           |                 |            |              |
| PIM SSM information for VRF: default      |           |                 |            |              |
| SSM group range : PIM_PLIST1              |           |                 |            |              |
| PIM SSM information for VRF: Vrf1         |           |                 |            |              |

```
SSM group range : PIM_PLIST1
```

```
sonic# show ip pim topology
PIM Multicast Routing Table for VRF: default

"Flags: S - Sparse, C - Connected, L - Local, P - Pruned,
R - RP-bit set, F - Register Flag, T - SPT-bit set, J - Join SPT,
K - Ack-Pending state"

(71.0.0.11, 233.0.0.1), uptime 13:08:24, expires 00:00:12, flags SCJT
Incoming interface: vlan100, RPF neighbor 100.0.0.1
Outgoing interface list:
 vlan200 uptime/expiry-time: 13:07:50/00:01:39
 vlan122 uptime/expiry-time: 12:33:21/---:---

(71.0.0.22, 233.0.0.1), uptime 13:08:45, expires 00:00:18, flags SCJT
Incoming interface: vlan100, RPF neighbor 100.0.0.1
Outgoing interface list:
 vlan200 uptime/expiry-time: 13:22:52/00:01:45
 vlan124 uptime/expiry-time: 12:42:28/---:---

(101.0.0.22, 225.1.1.1), uptime 13:07:51, expires 00:06:09, flags SCJT
Incoming interface: vlan105, RPF neighbor 105.0.0.1
Outgoing interface list:
 vlan200 uptime/expiry-time: 13:03:50/00:01:39
 vlan123 uptime/expiry-time: 13:02:40/---:---
```

```
sonic# show ip pim topology 233.0.0.1
PIM Multicast Routing Table for VRF: default
```

```
"Flags: S - Sparse, C - Connected, L - Local, P - Pruned,
R - RP-bit set, F - Register Flag, T - SPT-bit set, J - Join SPT,
K - Ack-Pending state"

(71.0.0.11, 233.0.0.1), uptime 13:08:24, expires 00:00:12, flags SCJT
Incoming interface: vlan100, RPF neighbor 100.0.0.1
Outgoing interface list:
 vlan200 uptime/expiry-time: 13:07:50/00:01:39
 vlan122 uptime/expiry-time: 12:33:21/---:---

(71.0.0.22, 233.0.0.1), uptime 13:08:45, expires 00:00:18, flags SCJT
Incoming interface: vlan100, RPF neighbor 100.0.0.1
Outgoing interface list:
 vlan200 uptime/expiry-time: 13:22:52/00:01:45
 vlan124 uptime/expiry-time: 12:42:28/---:---
```

```
sonic# show ip pim topology 225.1.1.1 101.0.0.22
PIM Multicast Routing Table for VRF: default
```

```
"Flags: S - Sparse, C - Connected, L - Local, P - Pruned,
R - RP-bit set, F - Register Flag, T - SPT-bit set, J - Join SPT,
K - Ack-Pending state"

(101.0.0.22, 225.1.1.1), uptime 13:07:51, expires 00:06:09, flags SCJT
Incoming interface: vlan105, RPF neighbor 105.0.0.1
Outgoing interface list:
 vlan200 uptime/expiry-time: 13:03:50/00:01:39
 vlan123 uptime/expiry-time: 13:02:40/---:---
```

```
sonic# show ip pim vrf Vrf1 topology
PIM Multicast Routing Table for VRF: Vrf1
```

```
"Flags: S - Sparse, C - Connected, L - Local, P - Pruned,
R - RP-bit set, F - Register Flag, T - SPT-bit set, J - Join SPT,
K - Ack-Pending state"
```

```
(51.0.0.11, 233.0.0.1), uptime 13:08:24, expires 00:00:12, flags SCJT
Incoming interface: vlan300, RPF neighbor 30.0.0.1
Outgoing interface list:
```

```
vlan301 uptime/expiry-time: 13:07:50/00:01:39
(51.0.0.22, 233.0.0.1), uptime 13:08:24, expires 00:00:12, flags SCJT
Incoming interface: vlan300, RPF neighbor 30.0.0.1
Outgoing interface list:
vlan301 uptime/expiry-time: 13:07:50/00:01:39
vlan302 uptime/expiry-time: 14:07:50/00:01:29
```

```
sonic# show ip pim vrf all topology
PIM Multicast Routing Table for VRF: default

"Flags: S - Sparse, C - Connected, L - Local, P - Pruned,
R - RP-bit set, F - Register Flag, T - SPT-bit set, J - Join SPT,
K - Ack-Pending state"

(71.0.0.11, 233.0.0.1), uptime 13:08:24, expires 00:00:12, flags SCJT
Incoming interface: vlan100, RPF neighbor 100.0.0.1
Outgoing interface list:
vlan200 uptime/expiry-time: 13:07:50/00:01:39
vlan122 uptime/expiry-time: 12:33:21/---:---

(71.0.0.22, 233.0.0.1), uptime 13:08:45, expires 00:00:18, flags SCJT
Incoming interface: vlan100, RPF neighbor 100.0.0.1
Outgoing interface list:
vlan200 uptime/expiry-time: 13:22:52/00:01:45
vlan124 uptime/expiry-time: 12:42:28/---:---

(101.0.0.22, 225.1.1.1), uptime 13:07:51, expires 00:06:09, flags SCJT
Incoming interface: vlan105, RPF neighbor 105.0.0.1
Outgoing interface list:
vlan200 uptime/expiry-time: 13:03:50/00:01:39
vlan123 uptime/expiry-time: 13:02:40/---:---

PIM Multicast Routing Table for VRF: Vrf1

"Flags: S - Sparse, C - Connected, L - Local, P - Pruned,
R - RP-bit set, F - Register Flag, T - SPT-bit set, J - Join SPT,
K - Ack-Pending state"

(51.0.0.11, 233.0.0.1), uptime 13:08:24, expires 00:00:12, flags SCJT
Incoming interface: vlan300, RPF neighbor 30.0.0.1
Outgoing interface list:
vlan301 uptime/expiry-time: 13:07:50/00:01:39

(51.0.0.22, 233.0.0.1), uptime 13:08:24, expires 00:00:12, flags SCJT
Incoming interface: vlan300, RPF neighbor 30.0.0.1
Outgoing interface list:
vlan301 uptime/expiry-time: 13:07:50/00:01:39
vlan302 uptime/expiry-time: 14:07:50/00:01:29
```

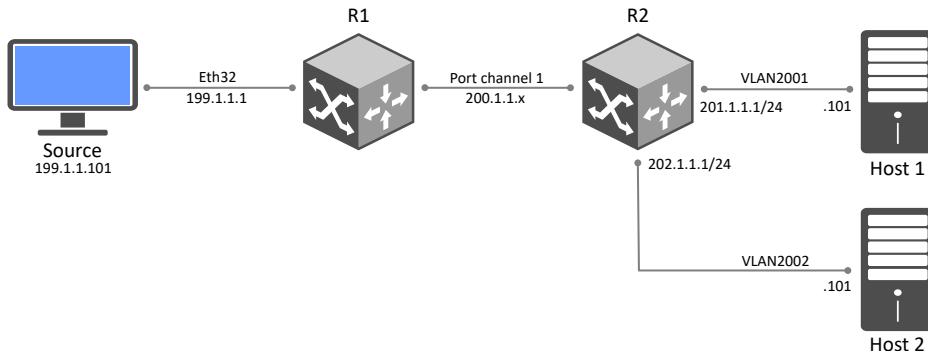
| Source<br>Pref | Group     | RpfInterface | RpfAddress | RibNextHop | Metric |
|----------------|-----------|--------------|------------|------------|--------|
| 71.0.0.11<br>1 | 233.0.0.1 | Vlan100      | 100.0.0.1  | 100.0.0.1  | 0      |
| 71.0.0.22<br>1 | 235.0.0.1 | Vlan100      | 100.0.0.1  | 100.0.0.1  | 0      |

| Source<br>Pref | Group     | RpfInterface | RpfAddress | RibNextHop | Metric |
|----------------|-----------|--------------|------------|------------|--------|
| 51.0.0.11<br>1 | 233.0.0.1 | Vlan300      | 30.0.0.1   | 30.0.0.1   | 0      |

| 51.0.0.22                            | 235.0.0.1 | Vlan300  | 30.0.0.1   | 30.0.0.1   | 0      |
|--------------------------------------|-----------|----------|------------|------------|--------|
| <hr/>                                |           |          |            |            |        |
| sonic# show ip pim vrf all rpf       |           |          |            |            |        |
| PIM RPF information for VRF: default |           |          |            |            |        |
| Source                               | Group     | RpfIface | RpfAddress | RibNextHop | Metric |
| 71.0.0.11                            | 233.0.0.1 | Vlan100  | 100.0.0.1  | 100.0.0.1  | 0      |
| 1                                    |           |          |            |            |        |
| 71.0.0.22                            | 235.0.0.1 | Vlan100  | 100.0.0.1  | 100.0.0.1  | 0      |
| 1                                    |           |          |            |            |        |
| PIM RPF information for VRF: Vrf1    |           |          |            |            |        |
| Source                               | Group     | RpfIface | RpfAddress | RibNextHop | Metric |
| 51.0.0.11                            | 233.0.0.1 | Vlan300  | 30.0.0.1   | 30.0.0.1   | 0      |
| 1                                    |           |          |            |            |        |
| 51.0.0.22                            | 235.0.0.1 | Vlan300  | 30.0.0.1   | 30.0.0.1   | 0      |
| 1                                    |           |          |            |            |        |

## Sample PIM-SSM configuration

This information describes how to enable PIM-SSM using the sample topology.



### Sample configuration on R1

```

R1#
R1# configure terminal
R1(config)# interface PortChannel 1
R1(conf-if-pol1)# ip address 200.1.1.2/24
R1(conf-if-pol1)# ip pim sparse-mode
R1(conf-if-pol1)# exit
R1(config)# router ospf
R1(config-router-ospf)# area 0
R1(config-router-ospf)# exit
R1(config)# interface PortChannel 1
R1(conf-if-pol1)# ip ospf area 0
R1(conf-if-pol1)# exit
R1(config)# interface Ethernet 56
R1(conf-if-Ethernet56)# no shutdown
R1(conf-if-Ethernet56)# channel-group 1
R1(conf-if-Ethernet56)# exit
R1(config)# interface Ethernet 32
R1(conf-if-Ethernet56)# no shutdown
R1(conf-if-Ethernet32)# ip address 199.1.1.1/24
R1(conf-if-Ethernet32)# ip pim sparse-mode
R1(conf-if-Ethernet32)# ip ospf area 0
R1(conf-if-Ethernet32)# exit

```

```

R1(config)# end
R1#
R1# configure terminal
R1(config)# ip prefix-list test_ssm seq 1 permit 225.1.0.0/16
R1(config)# ip pim ssm prefix-list test_ssm
R1(config)# end

```

### Sample configuration on R2

```

R2# configure terminal
R2(config)# interface Vlan 2001
R2(conf-if-Vlan2001)# ip address 201.1.1.1/24
R2(conf-if-Vlan2001)# ip pim sparse-mode
R2(conf-if-Vlan2001)# ip igmp
R2(conf-if-Vlan2001)# exit
R2(config)# interface Vlan 2002
R2(conf-if-Vlan2002)# ip address 202.1.1.1/24
R2(conf-if-Vlan2002)# ip pim sparse-mode
R2(conf-if-Vlan2002)# ip igmp
R2(conf-if-Vlan2002)# exit
R2(config)# interface Ethernet 32
R2(conf-if-Ethernet32)# no shutdown
R2(conf-if-Ethernet32)# switchport trunk allowed Vlan 2001
R2(conf-if-Ethernet32)# interface Ethernet 33
R2(conf-if-Ethernet33)# switchport trunk allowed Vlan 2002
R2(conf-if-Ethernet33)# no shutdown
R2(conf-if-Ethernet33)# end
R2# configure terminal
R2(config)# interface PortChannel 1
R2(conf-if-pol1)# ip address 200.1.1.1/24
R2(conf-if-pol1)# ip pim sparse-mode
R2(conf-if-pol1)# exit
R2(config)# router ospf
R2(config-router-ospf)# area 0
R2(config-router-ospf)# exit
R2(config)# interface PortChannel 1
R2(conf-if-pol1)# ip ospf area 0
R2(conf-if-pol1)# exit
R2(config)# interface Ethernet 56
R2(conf-if-Ethernet56)# no shutdown
R2(conf-if-Ethernet56)# channel-group 1
R2(conf-if-Ethernet56)# end

```

### Verify the PIM-SSM configuration on R1

The show ip pim interface command displays the PIM interface information.

| PIM interface information for VRF: default |       |           |          |           |             |                |
|--------------------------------------------|-------|-----------|----------|-----------|-------------|----------------|
| Interface                                  | State | Address   | PIM Nbrs | PIM DR    | DR-Priority | Hello-interval |
| Ethernet32<br>1                            | up    | 199.1.1.1 | 0        | 199.1.1.1 | 30          |                |
| PortChannel1<br>1                          | up    | 200.1.1.2 | 1        | 200.1.1.2 | 30          |                |

The show ip pim neighbor command displays the PIM neighbor information.

| PIM neighbor information for VRF: default |           |          |            |             |
|-------------------------------------------|-----------|----------|------------|-------------|
| Interface                                 | Neighbor  | Uptime   | Expirytime | DR-Priority |
| BFD-State<br>PortChannel1<br>-            | 200.1.1.1 | 00:13:38 | 00:01:36   | 1           |

| PIM multicast routing table for VRF: default                                                                                                           |  |  |  |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|
| Flags: S - Sparse, C - Connected, L - Local, P - Pruned,<br>R - RP-bit set, F - Register Flag, T - SPT-bit set, J - Join SPT,<br>K - Ack-Pending state |  |  |  |  |
| (199.1.1.101, 225.1.1.1), uptime 00:00:07, expires Never, flags SCJT                                                                                   |  |  |  |  |

```

Incoming interface: Ethernet32, RPF neighbor 199.1.1.101
Outgoing interface list:
 PortChannell1 uptime/expiry-time: 00:00:07/00:03:21

(199.1.1.101, 225.1.1.101), uptime 00:00:07, expires Never, flags SCJT
 Incoming interface: Ethernet32, RPF neighbor 199.1.1.101
 Outgoing interface list:
 PortChannell1 uptime/expiry-time: 00:00:07/00:03:21

```

```

R1# show ip pim rpf
PIM RPF information for VRF: default
Source Group RpfIface RpfAddress RpfNextHop
Metric Pref
199.1.1.101 225.1.1.1 Ethernet32 0.0.0.0 199.1.1.101
0 0
199.1.1.101 225.1.1.101 Ethernet32 0.0.0.0 199.1.1.101
0 0

```

The `show ip mroute` command displays IP multicast routes.

```

R1# show ip mroute
IP multicast routing table for VRF: default
 * -> indicates installed route
 Source Group Input Output
Uptime
* 199.1.1.101 225.1.1.1 Ethernet32 PortChannell1
00:00:18
* 199.1.1.101 225.1.1.101 Ethernet32 PortChannell1
00:00:18

```

### Verify the PIM-SSM configuration on R2

```

R2# show ip pim interface
PIM interface information for VRF: default
Interface State Address PIM Nbrs PIM DR Hello-interval
 PIM DR-Priority
PortChannell1 up 200.1.1.1 1 200.1.1.2 30
 1
Vlan2001 up 201.1.1.1 0 201.1.1.1 30
 1
Vlan2002 up 202.1.1.1 0 202.1.1.1 30
 1

```

```

R2# show ip pim neighbor
PIM neighbor information for VRF: default
Interface Neighbor Uptime Expirytime DR-Priority
BFD-State
PortChannell1 200.1.1.2 00:15:24 00:01:20 1
-

```

```

R2# show ip igmp groups
Interface Address Group Mode Timer Srcs V
 Uptime
Vlan2001 201.1.1.1 225.1.1.1 INCLUDE ---:---:--- 1 3
 00:00:32
Vlan2002 202.1.1.1 225.1.1.101 INCLUDE ---:---:--- 1 3
 00:00:32

```

```

R2# show ip igmp sources
Interface Address Group Source Timer Fwd
 Uptime
Vlan2001 201.1.1.1 225.1.1.1 199.1.1.101 04:02 Y
 00:00:36
Vlan2002 202.1.1.1 225.1.1.101 199.1.1.101 04:04 Y
 00:00:36

```

```

R2# show ip pim topology
PIM multicast routing table for VRF: default

```

```

Flags: S - Sparse, C - Connected, L - Local, P - Pruned,
R - RP-bit set, F - Register Flag, T - SPT-bit set, J - Join SPT,
K - Ack-Pending state
(199.1.1.101, 225.1.1.1), uptime 00:00:41, expires 00:00:56, flags SJT
 Incoming interface: PortChannel1, RPF neighbor 200.1.1.2
 Outgoing interface list:
 Vlan2001 uptime/expiry-time: --::--/Never

(199.1.1.101, 225.1.1.101), uptime 00:00:41, expires 00:00:56, flags SJT
 Incoming interface: PortChannel1, RPF neighbor 200.1.1.2
 Outgoing interface list:
 Vlan2002 uptime/expiry-time: --::--/Never

```

```

R2# show ip pim rpf
PIM RPF information for VRF: default
Source Group RpfIface RpfAddress RpfNextHop
Metric Pref
199.1.1.101 225.1.1.1 PortChannel1 200.1.1.2 200.1.1.2
20 110
199.1.1.101 225.1.1.101 PortChannel1 200.1.1.2 200.1.1.2
20 110

```

```

R2# show ip mroute
IP multicast routing table for VRF: default
 * -> indicates installed route
 Source Group Input Output
Uptime
* 199.1.1.101 225.1.1.1 PortChannel1 Vlan2001
00:01:09
* 199.1.1.101 225.1.1.101 PortChannel1 Vlan2002
00:01:09

```

```

R2# show ip mroute summary
IP multicast routing table summary for VRF: default
Mroute Type Installed/Total
(S,G) 2/2

```

## VXLAN

**i | NOTE:** VXLAN is available only in the Enterprise Standard, Enterprise Premium, and Edge Standard bundles. VXLAN is not available in the Cloud Standard and Cloud Premium bundles.

A virtual extensible LAN (VXLAN) extends Layer 2 (L2) server connectivity over an underlying Layer 3 (L3) transport network in a multi-tenant, virtualized data center. Enterprise SONiC supports VXLAN as described in RFC 7348.

VXLAN provides a L2 overlay mechanism on an existing L3 network by encapsulating (tunneling) L2 frames in L3 packets. The VXLAN-shared forwarding domain allows hosts, such as virtual and physical machines in tenant L2 segments, to communicate over the shared IP network. Each L2 tenant segment is identified by a 24-bit ID called a VXLAN network identifier (VNI). The unique VNI maintains L2 isolation (separate bridging domains) from other tenant segments.

When a switch is deployed as a VXLAN gateway, it performs encapsulation/de-encapsulation of L2 frames in L3 packets while tunneling server traffic. In this role, a switch operates as a VXLAN tunnel endpoint (VTEP). An IP routing protocol, such as BGP, provides IP reachability between VTEPs. Using VXLAN tunnels, VLAN server segments communicate through the extended L2 forwarding domain.

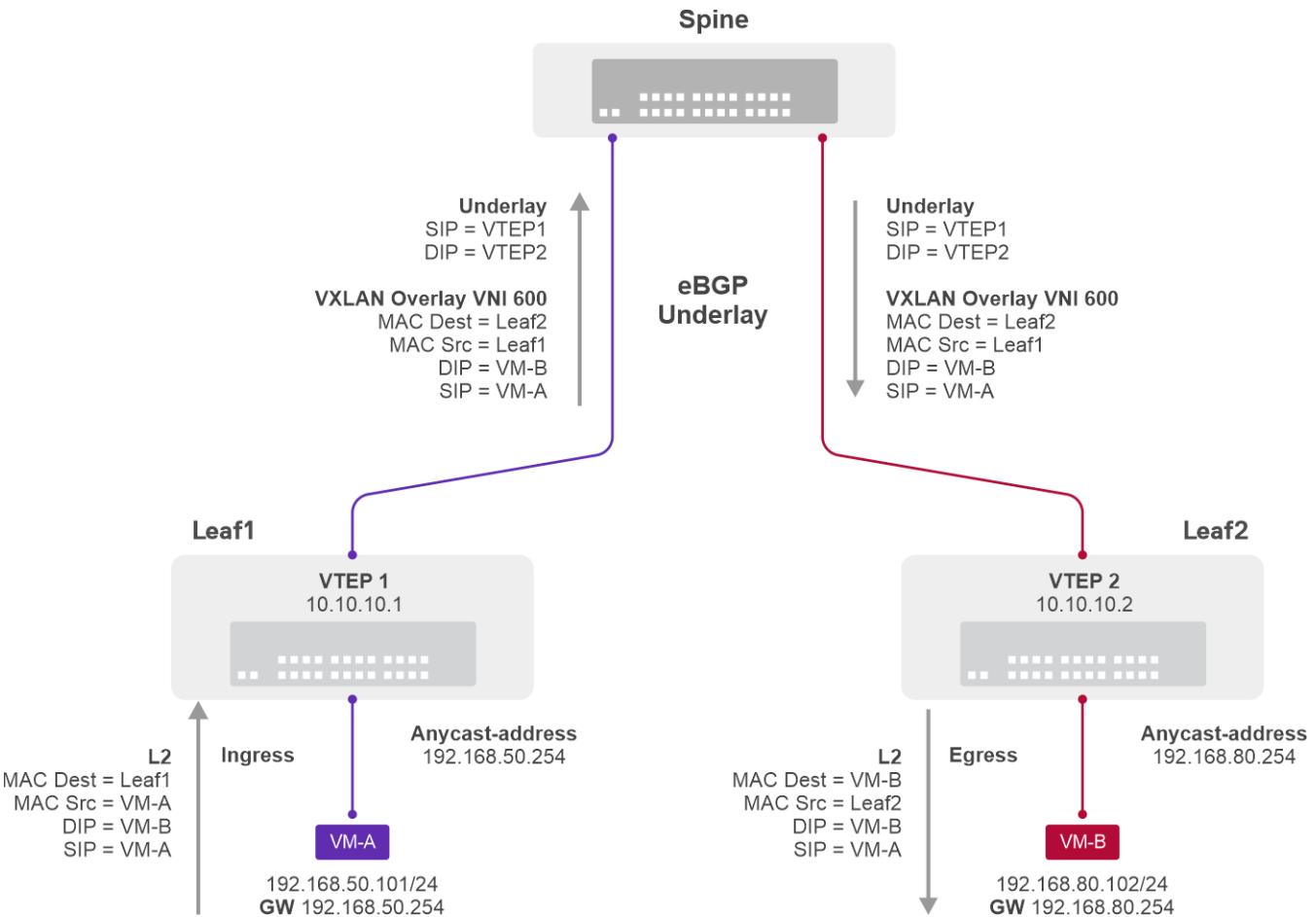
To extend L2 host subnets over an L3 network, VXLAN uses two types of integrated bridging and routing (IRB):

- In *symmetric* routing, all VTEPs can perform routing. Routing decisions are made on both ingress and egress VTEPs.
- In *asymmetric* routing, all VTEPs can perform routing. Routing decisions are made only on ingress VTEPs. Egress VTEPs only perform bridging.

### Symmetric IRB

In Symmetric IRB, both the ingress and egress VTEPs route a packet to its final destination. The ingress VTEP routes the packet to the final destination using the egress VTEP as the next hop. After the packet is decapsulated, the egress VTEP performs a routing lookup and routes it to the destination using the IP header. The VNI used to carry the packet between the ingress and egress VTEPs is a Layer 3 VNI. This Layer 3 VNI is different from the Layer 2 VNI of the source or destination network. In the following example, these actions are taken at each hop:

- VM-A sends a packet to VM-B using the Leaf1 anycast gateway for subnet 192.168.50.0. Both VM-A and VM-B are in the same VRF (Vrf1). The routing table for Vrf1 identifies reachability to subnet 192.168.80.0 through VTEP2 (10.10.10.2).
- VTEP1 encapsulates the packet for VM-A with a VXLAN header using the VTEP2 destination MAC address and VNI 600.
- The encapsulated packet is routed to the spine switch with the destination of the Leaf2 VTEP address. The spine switch routes the packet to the Leaf2 VTEP using the underlay routing table.
- The Leaf2 VTEP decapsulates the VXLAN header and routes the packet to VM-B.



**Figure 7. Symmetric IRB packet flow**

For more information, see [Configure symmetric IRB](#).

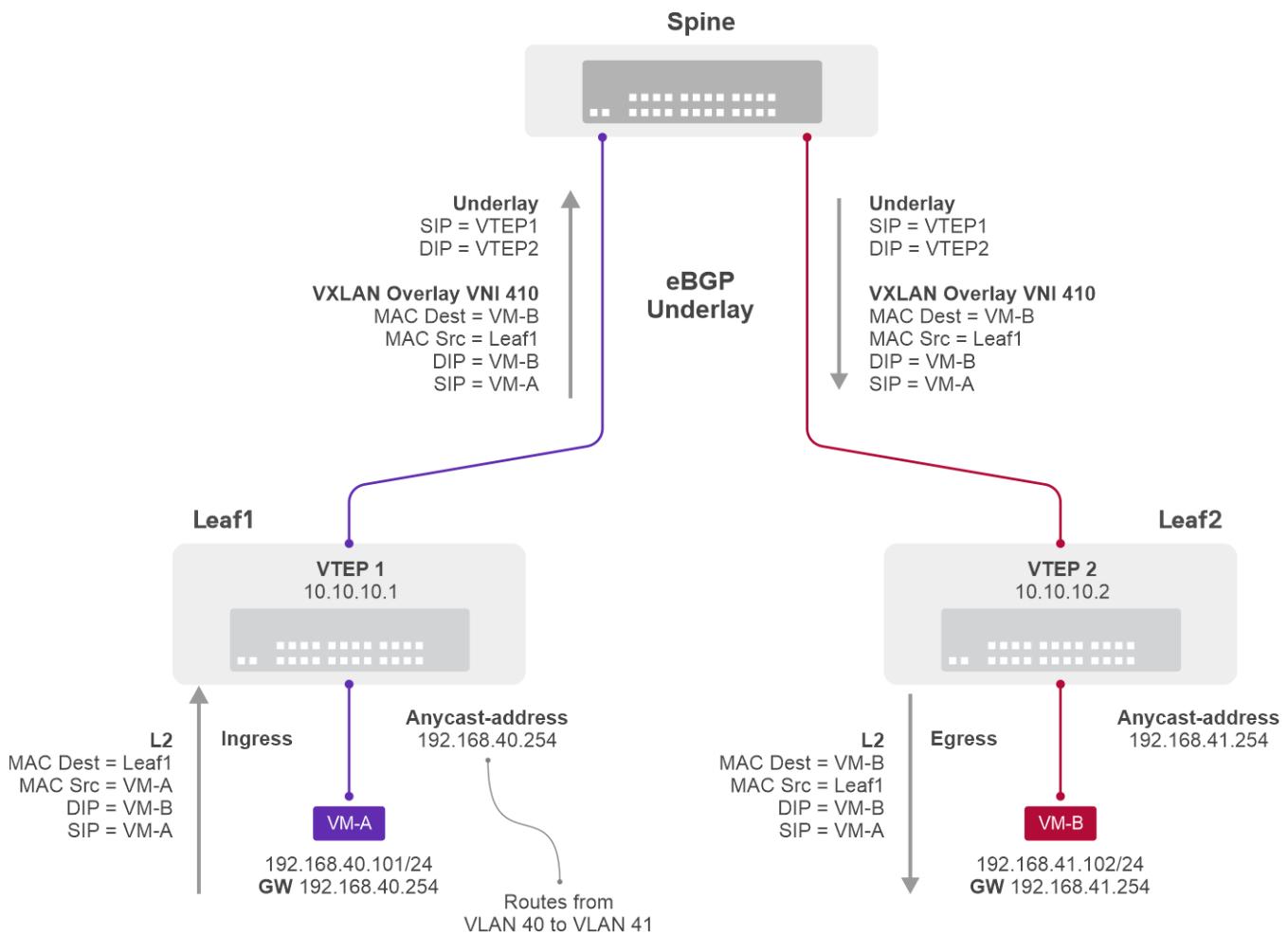
### Asymmetric IRB

In asymmetric IRB, each L2 host VLAN is mapped to a unique VXLAN VNI. If there is one tenant, the default VRF is used. Multiple L2 tenants use non-default VRFs. Ingress VTEPs perform routing and bridging; egress VTEPs perform only L2 bridging. Both originating and terminating VNIs must be configured on the originating and terminating VTEPs.

An ingress VTEP performs a routing lookup to decide to which egress VTEP a packet is sent. The egress VTEP decapsulates the packet and bridges it to its final destination using the MAC header.

To route and bridge packets, asymmetric IRB requires each ingress VTEP to store information about all tenant VLANs in the network. As a result, each VTEP uses more routing table memory (compared to symmetric IRB). For this reason, asymmetric IRB is usually deployed in small and medium-sized data centers. Because routing is performed only on ingress VTEPs, performance improves because the egress VTEP of a VXLAN tunnel performs only bridging, not routing as in symmetric IRB. In the following example, these actions are taken at each hop:

- VM-A sends a packet to VM-B using the Leaf1 anycast gateway for subnet 192.168.40.0. The destination IP address is found in the BGP Type-2 routing table and is associated with the VM-B MAC address, and VNI 410.
- VTEP 1 encapsulates the packet for VM-A with a VXLAN header using VNI 410, and the destination MAC and IP address for VM-B.
- The encapsulated packet is routed to the spine switch using the destination of the Leaf2 VTEP address. The spine switch routes the packet to the Leaf2 VTEP using the underlay routing table.
- The Leaf2 VTEP decapsulates the VXLAN header and bridges the packet to VM-B based on the destination MAC of the overlay.



**Figure 8. Asymmetric IRB packet flow**

For more information, see [Configure asymmetric IRB](#).

#### Symmetric and Asymmetric IRB comparison

**Table 35. Symmetric and Asymmetric IRB comparison**

| Symmetric IRB                                                                                                                           | Asymmetric IRB                                                                                                    |
|-----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Ingress VTEP needs only the MAC destination address of the corresponding egress VTEP router.                                            | Ingress VTEP needs the MAC destination address of the end station to which the packet is destined.                |
| Egress VTEP routes the packet after decapsulation.                                                                                      | Egress VTEP only bridges after decapsulation.                                                                     |
| Layer 3 VNI is mapped to VRFs and many Layer 2 VNIs are mapped to a specified VRF. Symmetric IRB cannot be deployed in the default VRF. | No Layer 3 VNI is required, however you can create VRFs for each tenant. Asymmetric IRB can be deployed in a VRF. |
| Centralized routing is not supported with symmetric IRB.                                                                                | Centralized routing is supported with asymmetric IRB.                                                             |

#### BUM storm control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. The traffic can be broadcast, unknown-unicast, or unknown-multicast (BUM). Enterprise SONiC supports a storm control feature that allows you to limit the amount of BUM traffic admitted to the system. To configure storm control on a port interface, use a `storm-control` command to specify the type of storm (broadcast, unknown-unicast or unknown multicast) and the maximum amount of broadcast packets in kilobits per second (kbps). Traffic that exceeds the configured rate is dropped. Unknown-multicast traffic consists of all multicast traffic that does not match any of the statically configured or dynamically learned multicast groups.

#### Topics:

- VXLAN concepts
- VXLAN as NVO solution
- Configure VXLAN
- Configure EVPN
- Multi-site data center interconnect
- EVPN multihoming

## VXLAN concepts

|                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Network virtualization overlay (NVO)</b> | An overlay network extends L2 connectivity between server virtual machines (VMs) in a tenant segment over an underlay L3 IP network. A tenant segment can be a group of hosts or servers that are spread across an underlay network. <ul style="list-style-type: none"> <li>• The NVO overlay network uses a separate L2 bridge domain (virtual network), which is independent of legacy VLAN forwarding.</li> <li>• The NVO underlay network operates in the default VRF using the existing L3 infrastructure and routing protocols.</li> </ul> |
| <b>Virtual extensible LAN (VXLAN)</b>       | A type of network virtualization overlay that encapsulates a tenant payload into IP UDP packets for transport across the IP underlay network.                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>VXLAN network identifier (VNI)</b>       | A 24-bit ID number that identifies a tenant segment and transmits in a VXLAN-encapsulated packet.                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>VXLAN tunnel endpoint (VTEP)</b>         | A switch with connected end hosts that are assigned to virtual networks. The virtual networks map to VXLAN segments. Local and remote VTEPs perform encapsulation and decapsulation of VXLAN headers for the traffic between end hosts. A VTEP is also known as a network virtualization edge (NVE) node.                                                                                                                                                                                                                                        |
| <b>Bridge domain</b>                        | A L2 domain that receives packets from member interfaces and forwards or floods them to other member interfaces based on the destination MAC address of the packet. <ul style="list-style-type: none"> <li>• Simple VLAN — A bridge domain that a VLAN ID represents. Traffic on all member ports is assigned with the same VLAN ID.</li> <li>• Virtual network — A bridge domain that a virtual network ID (VNI) represents. A virtual network supports overlay encapsulation and maps with a single VLAN ID.</li> </ul>                        |
| <b>Distributed routing</b>                  | All VTEPs in a virtual network perform intersubnet routing and serve as L3 gateways in two possible modes: <ul style="list-style-type: none"> <li>• Asymmetric routing — All VTEPs can perform routing. Routing decisions are made only on ingress VTEPs. Egress VTEPs perform bridging.</li> <li>• Symmetric routing — All VTEPs perform routing. Routing decisions are made on both ingress and egress VTEPs.</li> </ul>                                                                                                                       |
| <b>Host VLAN</b>                            | A VLAN with host device and virtual machine members that is mapped to a VNI. <ul style="list-style-type: none"> <li>• Only one VLAN ID can be mapped to a VNI.</li> <li>• VXLAN is ideally suited for existing tenant VLANs that stretch over an IP fabric.</li> </ul>                                                                                                                                                                                                                                                                           |

## VXLAN as NVO solution

Network virtualization overlay (NVO) is a solution that addresses the requirements of a multitenant data center, especially one with virtualized hosts. An NVO network is an overlay network that is used to extend L2 connectivity among VMs belonging to a tenant segment over an underlay IP network. Each tenant payload is encapsulated in an IP packet at the originating VTEP. To access the payload, the destination VTEP strips off the encapsulation. Each tenant segment is internally mapped to a virtual network ID (VNI).

VXLAN is a type of encapsulation that is used as an NVO solution. VXLAN encapsulates a tenant payload into IP UDP packets for transport across the IP underlay network. The VNI uniquely identifies the tenant segment on all VTEPs. ASIC tables:

- Enable creation of a L2 bridge flooding domain across a L3 network.
- Facilitate packet forwarding between local ports and tunneling packets from the local device to a remote device.

# Configure VXLAN

To extend L2 tenant segments by tunneling multiple L2 VLANs over a L3 underlay network, follow these configuration steps on each switch to set up a VXLAN:

1. Configure a switch to operate as a VXLAN tunnel endpoint (VTEP) for both symmetric and asymmetric IRB — see [Configure VTEP](#).
2. Configure a VXLAN to route L2 tenant traffic, and enable ARP suppression — see [Configure symmetric IRB](#) or [Configure asymmetric IRB](#). For symmetric IRB, configure a L3 VNI for tenant traffic.
3. Verify the VXLAN configuration — see [View symmetric IRB configuration](#) or [View asymmetric IRB configuration](#).
4. Configure BGP-based EVPN for VXLAN traffic — see [Configure EVPN](#).

## Configure VTEP

### Before you start

- Do not assign the source Loopback interface to a nondefault VRF instance. Do not assign the source IP address to end hosts in any VRF.
- Underlay reachability of remote tunnel endpoints is supported only in the default VRF.
- 

### VTEP Configuration

1. Configure a Loopback interface (0 to 16383) which will be used as the VTEP source IP address. The interface is automatically enabled and up.

```
sonic(config)# interface Loopback 1
sonic(conf-if-lo1)# ip address ip-address/mask
sonic(conf-if-lo1)# exit
```

2. Configure a source IP address using the Loopback IP address in Step 1. This VTEP source IP address is used in VXLAN-encapsulated packet headers. No source IP interface is configured by default. The VXLAN interface name is a text string; 10 characters maximum. Dell Technologies recommends that you use a VTEP name in the format `vtepnumber`, such as `vtep1`. Enter the source IP address in dotted decimal A.B.C.D format. An IPv6 address is not supported as the source VXLAN address. Only one source IP address is supported on a VTEP. On a stand-alone VTEP, the source IP address is used as both the source VXLAN tunnel endpoint in the overlay, and the IP source address for routing in the underlay.

**(i) NOTE:** If you use MLAG with a peer VTEP, configure the same source IP address on each VTEP peer. Peer VTEPs in an MLAG create a single *logical VTEP*.

```
sonic(config)# interface vxlan vxlan-interface-name
sonic(conf-if-vtep1)# source-ip {ipv4-address | Loopback number}
```

To delete a source IP address, enter the `no vxlan source-ip ipv4-address` command.

**(i) NOTE:** The VTEP is automatically assigned to an EVPN network virtualization overlay (NVO) network. The overlay network is a VXLAN domain that transports L2/L3 tenant traffic over an IP underlay.

3. Configure the anycast MAC address of the gateway used on all VTEPs to forward VXLAN traffic in the overlay. Configure the same anycast MAC address on each VTEP in the VXLAN network.

```
sonic(config)# ip anycast-address enable
sonic(config)# ip anycast-mac-address mac-address
```

4. (Optional) **MLAG peer in a logical VTEP:** If you use MLAG to configure two VTEPs to form a logical VTEP, it is recommended that you enable the advertisement of the primary IP address on each node.
  - Primary IP address configuration allows for optimal use of peer link bandwidth when you use orphan ports.
  - EVPN L3VNI (Type-5) routes are advertised with the primary IP address.
  - EVPN MAC/MAC-IP (Type-2) routes are advertised with the primary IP address for MAC addresses learned on orphan ports.
  - EVPN MAC/MAC-IP (Type-2) routes are advertised with the VTEP's source IP address for MAC addresses learned on MLAG client ports.
  - EVPN Inclusive Multicast Routes (Type-3) are always advertised with the VTEP source IP address.

To enable PIP address advertisement on a VTEP in an MLAG: Configure a primary IP address (in addition to the source IP address), create a unique loopback interface with the PIP address, then advertise the PIP address in the BGP default VRF; for example:

```
sonic(config)# interface vxlan vxlan-interface-name
sonic(conf-if-vtep1)# primary-ip 2.2.2.2

sonic(config)# interface Loopback 2
sonic(conf-if-lo2)# ip address 2.2.2.2/32
sonic(conf-if-lo2)# exit

sonic(config)# router bgp 10 vrf default
sonic(conf-router)# router-id 2.2.2.2
```

PIP address advertisement:

- Prevents the virtual IP address common to MLAG peers that form a logical VTEP from being advertised as the next-hop IP address.
- Allows EVPN L3VNI (Type-5) and Type-2 routes to be advertised with a VTEP address unique to each VTEP switch in the multi-chassis LAG/logical VTEP.

### VTEP configuration

```
sonic(config)# interface Loopback 1
sonic(conf-if-lo1)# ip address 10.10.10.25/32
sonic(conf-if-lo1)# exit

sonic(config)# interface vxlan vtep25
sonic(conf-if-vxlan-vtep25)# source-ip 10.10.10.25
sonic(conf-if-vxlan-vtep25)# exit

sonic(config)# ip anycast-address enable
sonic(config)# ip anycast-mac-address 00:00:00:01:02:03
```

## Configure symmetric IRB

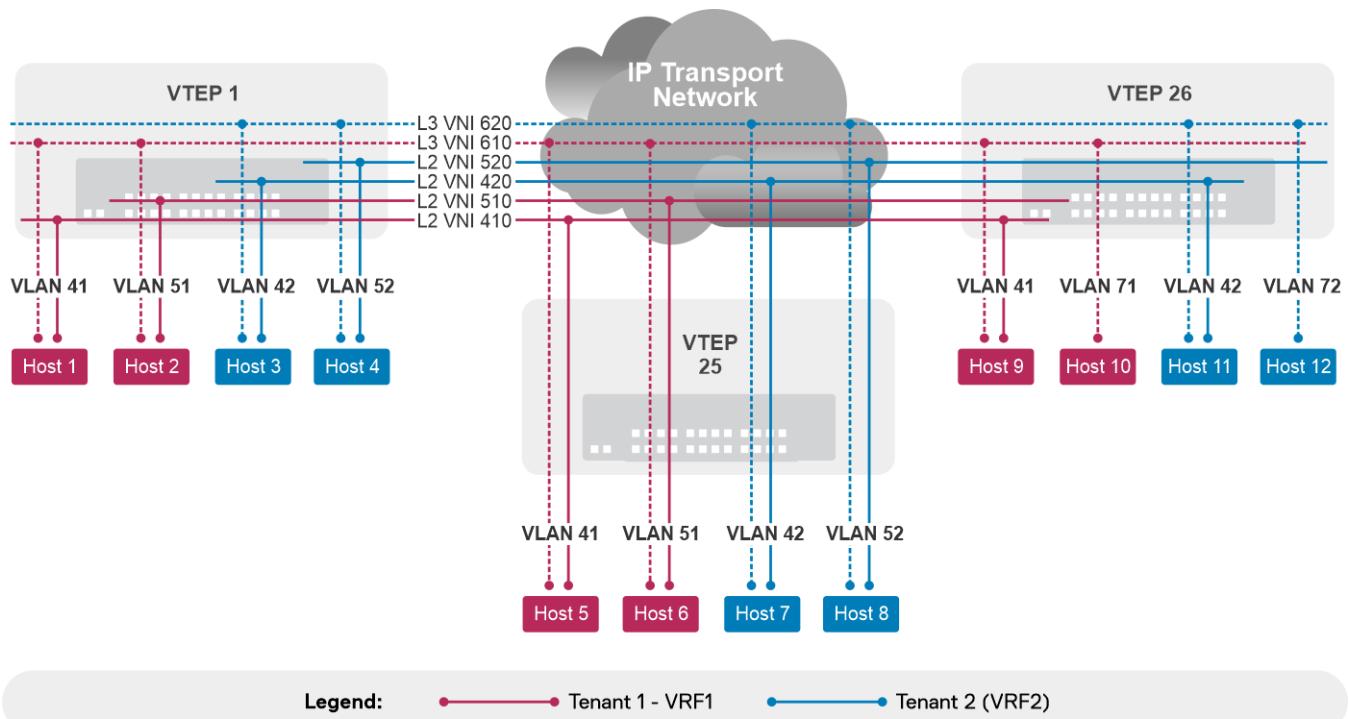


Figure 9. Symmetric IRB topology

Use symmetric IRB when your network requires symmetric routing and/or L3 multi-tenancy. With a L3 VXLAN, a tenant with VLANs in different subnets routes traffic between end hosts using a nondefault VRF. Each tenant uses its own unique VRF to segregate its traffic from the traffic of other tenants. In this way, the end hosts in virtual machines (VMs) of different tenants do not communicate with each other.

- The same L3 VXLAN VNI is used for all VLANs assigned to a tenant VRF.
- Each VTEP acts as the L3 gateway that allows a tenant's VLANs to communicate with each other.
- The VLANs in a tenant VRF route end-host traffic using symmetric IRB. In *symmetric* routing, all VTEPs perform routing on both ingress and egress VTEPs.

#### Before you start

- Configure the VTEP source IP address (see [Configure a VTEP](#)).
- Create dedicated tenant VRFs and assign tenant VLANs. L3 segments of VLAN hosts communicate through each tenant VRF.

#### Symmetric IRB configuration

1. Create a nondefault VRF instance for each L3 tenant segment. The VRF segregates L3 tenant hosts. A VRF name is a text string; 15 characters maximum. Dell Technologies recommends that you use a VRF name in the format `Vrfname`, such as `Vrfcustomer10`, `Vrf-1`, or `Vrf-blue`.

```
sonic(config) # ip vrf vrf-name
```

2. For each tenant, create a single, dedicated VLAN for overlay routing. Assign the tenant's VRF to the VLAN. It is not necessary to configure an IPv4 address.

```
sonic(config) # interface vlan vlan-id
sonic(conf-if-vlan) # ip vrf forwarding vrf-name
sonic(conf-if-vlan) # exit
```

3. Create a L3 VXLAN by mapping only the dedicated VLAN configured in Step 2 to a VXLAN VNI. Each host VLAN in the tenant VRF uses the VNI in a many-to-1 correspondence. Valid VNI numbers are 1 to 16777215. VLAN IDs are 1 to 4094. Reenter the command to configure additional VNI/dedicated VLAN mappings.

**(i) NOTE:** In a L3 VXLAN, each tenant has only its dedicated L3 VLAN mapped to a VNI. In a L2 VXLAN, each L2 host VLAN is mapped to a unique VNI.

```
sonic(config) # interface vxlan vxlan-interface-name
sonic(conf-if-vtep) # map vni vni vlan vlan-id
```

- `vni-id` — Identifies a VXLAN ID.
  - `vlan-id` — Identifies the dedicated L3 VLAN ID.
  - To delete a VLAN-VNI mapping, enter the `no vxlan map vni vni vlan vlan-id` command.
4. Map each L3 VNI to a nondefault tenant VRF. Reenter the command to configure additional tenant VRF/VNI mappings. The VRF segregates the routing of VLAN tenant traffic in different VXLAN subnets in the overlay.

```
sonic(config) # interface vxlan vxlan-interface-name
sonic(conf-if-vtep) # map vni vni vrf vrf-name
```

- `vrf vrf-name` — Identifies a nondefault tenant VRF on the switch that is created in Step 1.
  - `vni vni` — Identifies the L3 VNI associated with the VRF.
  - To delete a VRF-VNI mapping, enter the `no vxlan map vni vni vrf vrf-name` command.
5. Assign each tenant VRF to the server-facing VLANs that are associated with the tenant; for example:

```
sonic(config) # interface vlan 50
sonic(conf-if-vlan50) # ip vrf forwarding Vrf1
sonic(conf-if-vlan50) # ip anycast-address 192.168.50.254/24

sonic(config) # interface vlan 51
sonic(conf-if-vlan51) # ip vrf forwarding Vrf2
sonic(conf-if-vlan51) # ip anycast-address 192.168.51.254/24
```

**(i) NOTE:** In order to avoid VM mobility issues, you must configure the same anycast IP address for a specified server-facing VLAN on all the VTEPs on which the VLAN is extended.

## Example: Symmetric IRB configuration — Multiple tenants

### VTEP Leaf1:

```
! Create tenant VRFs
Leaf1(config)# ip vrf Vrf1
Leaf1(config)# ip vrf Vrf2

! Assign source VXLAN and Router ID addresses to loopback interfaces
Leaf1(config)# interface loopback 0
Leaf1(conf-if-lo0)# description Router-id
Leaf1(conf-if-lo0)# ip address 10.0.2.1/32
Leaf1(conf-if-lo0)# exit
Leaf1(config)# interface loopback 1
Leaf1(conf-if-lo1)# description Vtep
Leaf1(conf-if-lo1)# ip address 10.10.10.1/32
Leaf1(conf-if-lo1)# exit

! Configure host-facing VLANs for Tenant 1
Leaf1(config)# interface Vlan 41
Leaf1(conf-if-Vlan41)# ip vrf forwarding Vrf1
Leaf1(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf1(conf-if-Vlan41)# neigh-suppress
Leaf1(conf-if-Vlan41)# exit
Leaf1(config)# interface Vlan 51
Leaf1(conf-if-Vlan51)# ip vrf forwarding Vrf1
Leaf1(conf-if-Vlan51)# ip anycast-address 192.168.51.254/24
Leaf1(conf-if-Vlan51)# neigh-suppress
Leaf1(conf-if-Vlan51)# exit

! Configure host-facing VLANs for Tenant 2
Leaf1(config)# interface Vlan 42
Leaf1(conf-if-Vlan42)# ip vrf forwarding Vrf2
Leaf1(conf-if-Vlan42)# ip anycast-address 192.168.42.254/24
Leaf1(conf-if-Vlan42)# neigh-suppress
Leaf1(conf-if-Vlan42)# exit
Leaf1(config)# interface Vlan 52
Leaf1(conf-if-Vlan52)# ip vrf forwarding Vrf2
Leaf1(conf-if-Vlan52)# ip anycast-address 192.168.52.254/24
Leaf1(conf-if-Vlan52)# neigh-suppress
Leaf1(conf-if-Vlan52)# exit

! Assign VLANs to host interfaces
Leaf1(config)# interface Eth 1/1
Leaf1(conf-if-Eth1/1)# switchport trunk allowed vlan 41-42,51-52
Leaf1(conf-if-Eth1/1)# exit

Configure L3 VNI VLANs
Leaf1(config)# interface Vlan 61
Leaf1(conf-if-Vlan61)# ip vrf forwarding Vrf1
Leaf1(conf-if-Vlan61)# exit
Leaf1(config)# interface Vlan 62
Leaf1(conf-if-Vlan62)# ip vrf forwarding Vrf2
Leaf1(conf-if-Vlan62)# exit

! Map VNIs to VLANs and L3 VNIs to VRFs
Leaf1(config)# interface vxlan vtep-1
Leaf1(conf-if-vxlan-vtep-1)# source-ip 10.10.10.1
Leaf1(conf-if-vxlan-vtep-1)# map vni 410 vlan 41
Leaf1(conf-if-vxlan-vtep-1)# map vni 420 vlan 42
Leaf1(conf-if-vxlan-vtep-1)# map vni 510 vlan 51
Leaf1(conf-if-vxlan-vtep-1)# map vni 520 vlan 52
Leaf1(conf-if-vxlan-vtep-1)# map vni 610 vlan 61
Leaf1(conf-if-vxlan-vtep-1)# map vni 620 vlan 62
Leaf1(conf-if-vxlan-vtep-1)# map vni 610 vrf Vrf1
Leaf1(conf-if-vxlan-vtep-1)# map vni 620 vrf Vrf2
Leaf1(conf-if-vxlan-vtep-1)# exit
```

**VTEP Leaf25:**

```
! Create tenant VRFs
Leaf25(config)# ip vrf Vrf1
Leaf25(config)# ip vrf Vrf2

! Assign source VXLAN and Router ID addresses to loopback interfaces
Leaf25(config)# interface loopback 0
Leaf25(conf-if-lo0)# description Router-id
Leaf25(conf-if-lo0)# ip address 10.0.2.25/32
Leaf25(conf-if-lo0)# exit
Leaf25(config)# interface loopback 1
Leaf25(conf-if-lo1)# description Vtep
Leaf25(conf-if-lo1)# ip address 10.10.10.25/32
Leaf25(conf-if-lo1)# exit

! Configure host-facing VLANs for Tenant 1
Leaf25(config)# interface Vlan 41
Leaf25(conf-if-Vlan41)# ip vrf forwarding Vrf1
Leaf25(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf25(conf-if-Vlan41)# neigh-suppress
Leaf25(conf-if-Vlan41)# exit
Leaf25(config)# interface Vlan 51
Leaf25(conf-if-Vlan51)# ip vrf forwarding Vrf1
Leaf25(conf-if-Vlan51)# ip anycast-address 192.168.51.254/24
Leaf25(conf-if-Vlan51)# neigh-suppress
Leaf25(conf-if-Vlan51)# exit

! Configure host-facing VLANs for Tenant 2
Leaf25(config)# interface Vlan 42
Leaf25(conf-if-Vlan42)# ip vrf forwarding Vrf2
Leaf25(conf-if-Vlan42)# ip anycast-address 192.168.42.254/24
Leaf25(conf-if-Vlan42)# neigh-suppress
Leaf25(conf-if-Vlan42)# exit
Leaf25(config)# interface Vlan 52
Leaf25(conf-if-Vlan52)# ip vrf forwarding Vrf2
Leaf25(conf-if-Vlan52)# ip anycast-address 192.168.52.254/24
Leaf25(conf-if-Vlan52)# neigh-suppress
Leaf25(conf-if-Vlan52)# exit

! Assign VLANs to host interfaces
Leaf25(config)# interface Eth 1/1
Leaf25(conf-if-Eth1/1)# switchport trunk allowed vlan 41-42,51-52
Leaf25(conf-if-Eth1/1)# exit

Configure L3 VNI VLANs
Leaf25(config)# interface Vlan 61
Leaf25(conf-if-Vlan61)# ip vrf forwarding Vrf1
Leaf25(conf-if-Vlan61)# exit
Leaf25(config)# interface Vlan 62
Leaf25(conf-if-Vlan62)# ip vrf forwarding Vrf2
Leaf25(conf-if-Vlan62)# exit

! Map VNIs to VLANs and L3 VNIs to VRFs
Leaf25(config)# interface vxlan vtep-25
Leaf25(conf-if-vxlan-vtep-25)# source-ip 10.10.10.1
Leaf25(conf-if-vxlan-vtep-25)# map vni 410 vlan 41
Leaf25(conf-if-vxlan-vtep-25)# map vni 420 vlan 42
Leaf25(conf-if-vxlan-vtep-25)# map vni 510 vlan 51
Leaf25(conf-if-vxlan-vtep-25)# map vni 520 vlan 52
Leaf25(conf-if-vxlan-vtep-25)# map vni 610 vlan 61
Leaf25(conf-if-vxlan-vtep-25)# map vni 620 vlan 62
Leaf25(conf-if-vxlan-vtep-25)# map vni 610 vrf Vrf1
Leaf25(conf-if-vxlan-vtep-25)# map vni 620 vrf Vrf2
Leaf25(conf-if-vxlan-vtep-25)# exit
```

**VTEP Leaf26:**

```
! Create tenant VRFs
Leaf26(config)# ip vrf Vrf1
Leaf26(config)# ip vrf Vrf2
```

```

! Assign source VXLAN and Router ID addresses to loopback interfaces
Leaf26(config)# interface loopback 0
Leaf26(conf-if-lo0)# description Router-id
Leaf26(conf-if-lo0)# ip address 10.0.2.26/32
Leaf26(conf-if-lo0)# exit
Leaf26(config)# interface loopback 1
Leaf26(conf-if-lo1)# description Vtep
Leaf26(conf-if-lo1)# ip address 10.10.10.26/32
Leaf26(conf-if-lo1)# exit

! Configure host-facing VLANs for Tenant 1
Leaf26(config)# interface Vlan 41
Leaf26(conf-if-Vlan41)# ip vrf forwarding Vrf1
Leaf26(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf26(conf-if-Vlan41)# neigh-suppress
Leaf26(conf-if-Vlan41)# exit
Leaf26(config)# interface Vlan 71
Leaf26(conf-if-Vlan71)# ip vrf forwarding Vrf1
Leaf26(conf-if-Vlan71)# ip anycast-address 192.168.71.254/24
Leaf26(conf-if-Vlan71)# neigh-suppress
Leaf26(conf-if-Vlan71)# exit

! Configure host-facing VLANs for Tenant 2
Leaf26(config)# interface Vlan 42
Leaf26(conf-if-Vlan42)# ip vrf forwarding Vrf2
Leaf26(conf-if-Vlan42)# ip anycast-address 192.168.42.254/24
Leaf26(conf-if-Vlan42)# neigh-suppress
Leaf26(conf-if-Vlan42)# exit
Leaf26(config)# interface Vlan 72
Leaf26(conf-if-Vlan72)# ip vrf forwarding Vrf2
Leaf26(conf-if-Vlan72)# ip anycast-address 192.168.72.254/24
Leaf26(conf-if-Vlan72)# neigh-suppress
Leaf26(conf-if-Vlan72)# exit

! Assign VLANs to host interfaces
Leaf26(config)# interface Eth 1/1
Leaf26(conf-if-Eth1/1)# switchport trunk allowed vlan 41-42,71-72
Leaf26(conf-if-Eth1/1)# exit

Configure L3 VNI VLANs
Leaf26(config)# interface Vlan 61
Leaf26(conf-if-Vlan61)# ip vrf forwarding Vrf1
Leaf26(conf-if-Vlan61)# exit
Leaf26(config)# interface Vlan 62
Leaf26(conf-if-Vlan62)# ip vrf forwarding Vrf2
Leaf26(conf-if-Vlan62)# exit

! Map VNIs to VLANs and L3 VNIs to VRFs
Leaf26(config)# interface vxlan vtep-26
Leaf26(conf-if-vxlan-vtep-26)# source-ip 10.10.10.26
Leaf26(conf-if-vxlan-vtep-26)# map vni 410 vlan 41
Leaf26(conf-if-vxlan-vtep-26)# map vni 420 vlan 42
Leaf26(conf-if-vxlan-vtep-26)# map vni 610 vlan 61
Leaf26(conf-if-vxlan-vtep-26)# map vni 620 vlan 62
Leaf26(conf-if-vxlan-vtep-26)# map vni 610 vrf Vrf1
Leaf26(conf-if-vxlan-vtep-26)# map vni 620 vrf Vrf2
Leaf26(conf-if-vxlan-vtep-26)# exit

```

## View symmetric IRB configuration

### View symmetric IRB configuration — Multiple tenants

```

Leaf1# show vxlan interface
VTEP Name : vtep-1
VTEP Source IP : 10.10.10.1
EVPN NVO Name : nvol
EVPN VTEP : vtep-1

```

```
Source Interface : Loopback1
PrimaryIP Interface : Not Configured
```

```
Leaf1# show vxlan vlandnimap
VLAN VNI
===== =====
Vlan41 410
Vlan42 420
Vlan51 510
Vlan52 520
Vlan61 610
Vlan62 620
Total count : 6
```

```
Leaf1# show vxlan vrfvnimap
VRF VNI
===== =====
Vrf1 610
Vrf2 620
Total count : 2
```

```
Leaf1# show neighbor-suppress-status

VlanId SuppressionStatus

Vlan41 on
Vlan42 on
Vlan51 on
Vlan52 on
Vlan61 off
Vlan62 off
```

## Configure asymmetric IRB

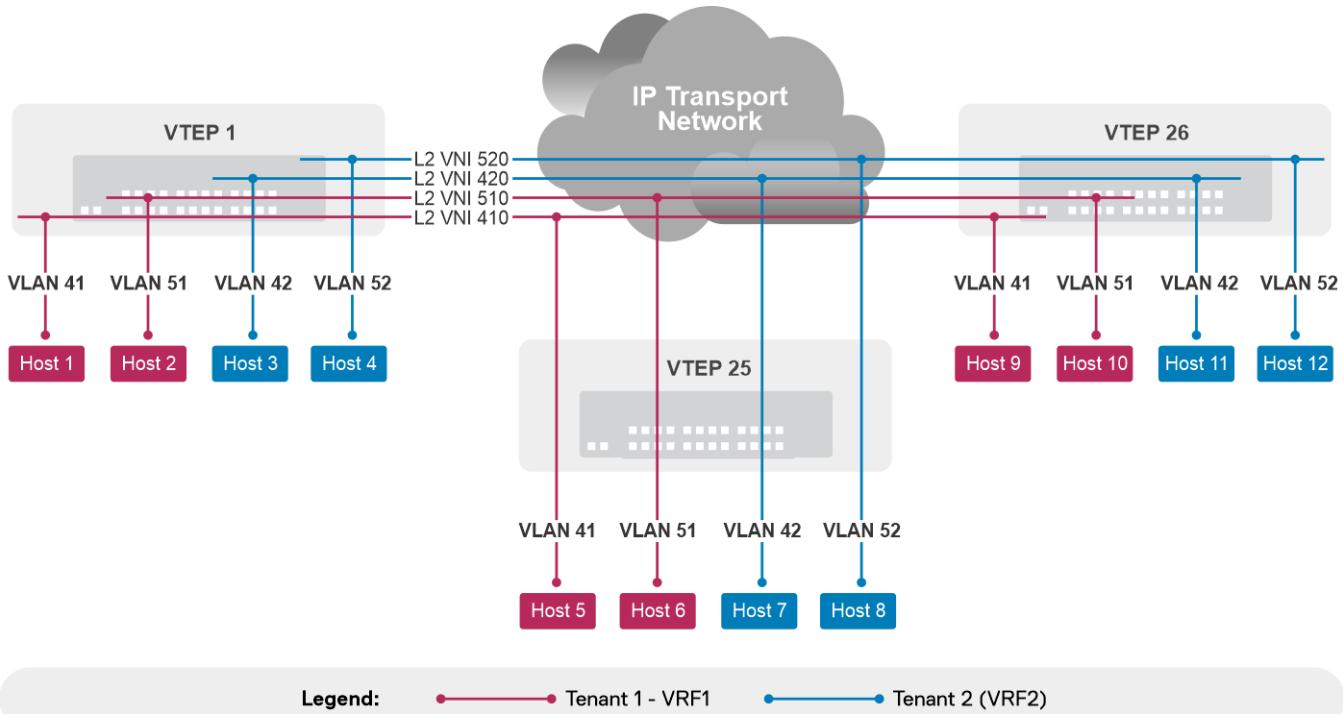


Figure 10. Asymmetric IRB topology

Configure asymmetric IRB in a simple VXLAN topology in which routing decisions are made only on ingress VTEPs. Egress VTEPs only perform bridging. In symmetric IRB all VTEPs also perform routing, but routing decisions are made on both ingress and egress VTEPs.

#### Before you start

- Configure the VTEP source IP address (see [Configure a VTEP](#)).
- Create the VLANs through which L2 segments of hosts communicate.

#### Asymmetric IRB configuration

- Configure the association between a L2 host VLAN and a VXLAN segment. Each VLAN maps to a single VNI in a 1:1 correspondence. In a VXLAN, VLAN packets are encapsulated with the VNI in the packet header. Enter this command to map VNIs to VLANs. Valid VNI numbers are 1 to 16777215. VLAN IDs are 1 to 4094.

```
sonic(config)# interface vxlan vxlan-interface-name
sonic(conf-if-vtep)# map vni vni-start vlan vlan-id-start [count n]
```

- vni-id-start* — Identifies a VXLAN ID.
- vlan-id-start* — Identifies a L2 VLAN ID.
- To configure the mapping of multiple contiguous VLAN IDs to multiple contiguous VXLAN IDs:
  - Enter the first VXLAN VNI in *vni-id-start*.
  - Enter the first VLAN ID in *vlan-id-start*.
  - (Optional) Enter the number of contiguous VLAN-VNI mappings to configure in *count n*. By default, *n* is 1.

For example, the following command creates 200 contiguous VLAN/VNI mappings: VLAN 100/VNI 1200, VLAN 101/VNI 1201 to VLAN 299/VNI1399.

```
sonic(config)# interface vxlan vtep-1
sonic(conf-if-vxlan-vtep-1)# map vni 1200 vlan 100 count 200
```

- To delete a single or contiguous VLAN-VNI mapping, enter the `no map vni vni-start vlan vlan-id-start count [n]` command.
- Configure an anycast IP address that is used as the gateway IP address for VMs on a tenant VLAN. The common anycast IP address supports seamless migration of hosts and virtual machines on different VTEPs.

```
sonic(config)# interface Vlan vlan-id
sonic(conf-if-Vlan40)# ip anycast-address ip-address/mask
```

- Enable ARP suppression. Using ARP suppression, VXLAN can reduce network flooding that is caused by broadcast traffic from ARP requests. When you enable ARP suppression for a L2 tenant VLAN, the VTEP maintains a table of MAC/IP bindings for the associated VNI. When a VLAN end host in the VNI sends an ARP request for another end-host IP address, the local VTEP can reply with the IP address in its ARP suppression table.

```
sonic(config)# interface vlan vlan-id
sonic(conf-if-vlan)# neigh-suppress
```

- (Optional) For multiple tenants, create a VRF for each tenant. Assign a tenant's host VLANs to the VRF. Configure each host VLAN with an IP anycast address. Map each host VLAN to a unique VXLAN VNI. For example, to add a tenant to a VXLAN using a nondefault VRF:

```
! Create a unique tenant VRF
sonic(config)# ip vrf Vrf4

! Configure host VLAN 140 and assign it to the tenant VRF
sonic(config)# interface vlan 140
sonic(conf-if-Vlan140)# ip vrf forwarding Vrf4
sonic(conf-if-Vlan140)# ip anycast-address 192.168.140.254/24
sonic(conf-if-Vlan140)# neigh-suppress
sonic(conf-if-Vlan140)# exit
sonic(config)#

! Configure host VLAN 141 and assign it to the tenant VRF
sonic(config)# interface vlan 141
sonic(conf-if-Vlan141)# ip vrf forwarding Vrf4
sonic(conf-if-Vlan141)# ip anycast-address 192.168.141.254/24
sonic(conf-if-Vlan141)# neigh-suppress
sonic(conf-if-Vlan141)# exit
sonic(config) #
```

```

! Map host VLANs on the VTEP to VNIs
sonic(config)# interface vxlan vtep-1
sonic(conf-if-vxlan-vtep-1)# map vni 1400 vlan 140
sonic(conf-if-vxlan-vtep-1)# map vni 1410 vlan 141

```

## Example: Asymmetric IRB configuration — Multiple tenants

### VTEP Leaf1:

```

! Create tenant VRFs
Leaf1(config)# ip vrf Vrf1
Leaf1(config)# ip vrf Vrf2

! Assign source VXLAN and Router ID addresses to loopback interfaces
Leaf1(config)# interface loopback 0
Leaf1(conf-if-lo0)# description Router-id
Leaf1(conf-if-lo0)# ip address 10.0.2.1/32
Leaf1(conf-if-lo0)# exit
Leaf1(config)# interface loopback 1
Leaf1(conf-if-lo1)# description Vtep
Leaf1(conf-if-lo1)# ip address 10.10.10.1/32
Leaf1(conf-if-lo1)# exit

! Configure host-facing VLANs for Tenant 1
Leaf1(config)# interface Vlan 41
Leaf1(conf-if-Vlan41)# ip vrf forwarding Vrf1
Leaf1(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf1(conf-if-Vlan41)# neigh-suppress
Leaf1(conf-if-Vlan41)# exit
Leaf1(config)# interface Vlan 51
Leaf1(conf-if-Vlan51)# ip vrf forwarding Vrf1
Leaf1(conf-if-Vlan51)# ip anycast-address 192.168.51.254/24
Leaf1(conf-if-Vlan51)# neigh-suppress
Leaf1(conf-if-Vlan51)# exit

! Configure host-facing VLANs for Tenant 2
Leaf1(config)# interface Vlan 42
Leaf1(conf-if-Vlan42)# ip vrf forwarding Vrf2
Leaf1(conf-if-Vlan42)# ip anycast-address 192.168.42.254/24
Leaf1(conf-if-Vlan42)# neigh-suppress
Leaf1(conf-if-Vlan42)# exit
Leaf1(config)# interface Vlan 52
Leaf1(conf-if-Vlan52)# ip vrf forwarding Vrf2
Leaf1(conf-if-Vlan52)# ip anycast-address 192.168.52.254/24
Leaf1(conf-if-Vlan52)# neigh-suppress
Leaf1(conf-if-Vlan52)# exit

! Assign VLANs to host interfaces
Leaf1(config)# interface Eth 1/1
Leaf1(conf-if-Eth1/1)# switchport trunk allowed vlan 41-42,51-52
Leaf1(conf-if-Eth1/1)# exit

! Map VNIs to VLANs
Leaf1(config)# interface vxlan vtep-1
Leaf1(conf-if-vxlan-vtep-1)# source-ip 10.10.10.1
Leaf1(conf-if-vxlan-vtep-1)# map vni 410 vlan 41
Leaf1(conf-if-vxlan-vtep-1)# map vni 420 vlan 42
Leaf1(conf-if-vxlan-vtep-1)# map vni 510 vlan 51
Leaf1(conf-if-vxlan-vtep-1)# map vni 520 vlan 52
Leaf1(conf-if-vxlan-vtep-1)# exit

```

### VTEP Leaf25:

```

! Create tenant VRFs
Leaf25(config)# ip vrf Vrf1
Leaf25(config)# ip vrf Vrf2

! Assign source VXLAN and Router ID addresses to loopback interfaces
Leaf25(config)# interface loopback 0
Leaf25(conf-if-lo0)# description Router-id

```

```

Leaf25(conf-if-lo0)# ip address 10.0.2.25/32
Leaf25(conf-if-lo0)# exit
Leaf25(config)# interface loopback 1
Leaf25(conf-if-lo1)# description Vtep
Leaf25(conf-if-lo1)# ip address 10.10.10.25/32
Leaf25(conf-if-lo1)# exit

! Configure host-facing VLANs for Tenant 1
Leaf25(config)# interface Vlan 41
Leaf25(conf-if-Vlan41)# ip vrf forwarding Vrf1
Leaf25(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf25(conf-if-Vlan41)# neigh-suppress
Leaf25(conf-if-Vlan41)# exit
Leaf25(config)# interface Vlan 51
Leaf25(conf-if-Vlan51)# ip vrf forwarding Vrf1
Leaf25(conf-if-Vlan51)# ip anycast-address 192.168.51.254/24
Leaf25(conf-if-Vlan51)# neigh-suppress
Leaf25(conf-if-Vlan51)# exit

! Configure host-facing VLANs for Tenant 2
Leaf25(config)# interface Vlan 42
Leaf25(conf-if-Vlan42)# ip vrf forwarding Vrf2
Leaf25(conf-if-Vlan42)# ip anycast-address 192.168.42.254/24
Leaf25(conf-if-Vlan42)# neigh-suppress
Leaf25(conf-if-Vlan42)# exit
Leaf25(config)# interface Vlan 52
Leaf25(conf-if-Vlan52)# ip vrf forwarding Vrf2
Leaf25(conf-if-Vlan52)# ip anycast-address 192.168.52.254/24
Leaf25(conf-if-Vlan52)# neigh-suppress
Leaf25(conf-if-Vlan52)# exit

! Assign VLANs to host interfaces
Leaf25(config)# interface Eth 1/1
Leaf25(conf-if-Eth1/1)# switchport trunk allowed vlan 41-42,51-52
Leaf25(conf-if-Eth1/1)# exit

! Map VNIs to VLANs
Leaf25(config)# interface vxlan vtep-25
Leaf25(conf-if-vxlan-vtep-25)# source-ip 10.10.10.25/32
Leaf25(conf-if-vxlan-vtep-25)# map vni 410 vlan 41
Leaf25(conf-if-vxlan-vtep-25)# map vni 420 vlan 42
Leaf25(conf-if-vxlan-vtep-25)# map vni 510 vlan 51
Leaf25(conf-if-vxlan-vtep-25)# map vni 520 vlan 52
Leaf25(conf-if-vxlan-vtep-25)# exit

```

#### VTEP Leaf26:

```

! Create tenant VRFs
Leaf26(config)# ip vrf Vrf1
Leaf26(config)# ip vrf Vrf2

! Assign source VXLAN and Router ID addresses to loopback interfaces
Leaf26(config)# interface loopback 0
Leaf26(conf-if-lo0)# description Router-id
Leaf26(conf-if-lo0)# ip address 10.0.2.26/32
Leaf26(conf-if-lo0)# exit
Leaf26(config)# interface loopback 1
Leaf26(conf-if-lo1)# description Vtep
Leaf26(conf-if-lo1)# ip address 10.10.10.26/32
Leaf26(conf-if-lo1)# exit

! Configure host-facing VLANs for Tenant 1
Leaf26(config)# interface Vlan 41
Leaf26(conf-if-Vlan41)# ip vrf forwarding Vrf1
Leaf26(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf26(conf-if-Vlan41)# neigh-suppress
Leaf26(conf-if-Vlan41)# exit
Leaf26(config)# interface Vlan 51
Leaf26(conf-if-Vlan51)# ip vrf forwarding Vrf1
Leaf26(conf-if-Vlan51)# ip anycast-address 192.168.51.254/24
Leaf26(conf-if-Vlan51)# neigh-suppress
Leaf26(conf-if-Vlan51)# exit

```

```

! Configure host-facing VLANs for Tenant 2
Leaf26(config)# interface Vlan 42
Leaf26(conf-if-Vlan42)# ip vrf forwarding Vrf2
Leaf26(conf-if-Vlan42)# ip anycast-address 192.168.42.254/24
Leaf26(conf-if-Vlan42)# neigh-suppress
Leaf26(conf-if-Vlan42)# exit
Leaf26(config)# interface Vlan 52
Leaf26(conf-if-Vlan52)# ip vrf forwarding Vrf2
Leaf26(conf-if-Vlan52)# ip anycast-address 192.168.52.254/24
Leaf26(conf-if-Vlan52)# neigh-suppress
Leaf26(conf-if-Vlan52)# exit

! Assign VLANs to host interfaces
Leaf26(config)# interface Eth 1/1
Leaf26(conf-if-Eth1/1)# switchport trunk allowed vlan 41-42,71-72
Leaf26(conf-if-Eth1/1)# exit

! Map VNIs to VLANs
Leaf26(config)# interface vxlan vtep-26
Leaf26(conf-if-vxlan-vtep-26)# source-ip 10.10.10.26
Leaf26(conf-if-vxlan-vtep-26)# map vni 410 vlan 41
Leaf26(conf-if-vxlan-vtep-26)# map vni 420 vlan 42
Leaf26(conf-if-vxlan-vtep-26)# map vni 510 vlan 51
Leaf26(conf-if-vxlan-vtep-26)# map vni 520 vlan 52
Leaf26(conf-if-vxlan-vtep-26)# exit

```

## Example: Asymmetric IRB configuration — Default VRF

### VTEP Leaf1:

```

Leaf1(config)# interface Loopback 1
Leaf1(conf-if-lol1)# description VTEP
Leaf1(conf-if-lol1)# ip address 10.10.10.1/32
Leaf1(conf-if-lol1)# exit

Leaf1(config)# ip anycast-address enable
Leaf1(config)# ip anycast-mac-address 00:00:00:00:01:02

Leaf1(config)# interface Vlan 40
Leaf1(conf-if-Vlan40)# ip anycast-address 192.168.40.254/24
Leaf1(conf-if-Vlan40)# neigh-suppress
Leaf1(conf-if-Vlan40)# exit

Leaf1(config)# interface Vlan 41
Leaf1(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf1(conf-if-Vlan41)# neigh-suppress
Leaf1(conf-if-Vlan41)# exit

Leaf1(config)# interface PortChannel 100
Leaf1(conf-if-po100)# switchport trunk allowed vlan add 40,41
Leaf1(conf-if-po100)# exit

Leaf1(config)# interface PortChannel 202
Leaf1(conf-if-po202)# switchport trunk allowed vlan add 40,41
Leaf1(conf-if-po202)# exit

Leaf1(config)# interface vxlan vtep1
Leaf1(conf-if-vxlan-vtep-1)# source-ip 10.10.10.1
Leaf1(conf-if-vxlan-vtep-1)# map vni 400 vlan 40
Leaf1(conf-if-vxlan-vtep-1)# map vni 410 vlan 41
Leaf1(conf-if-vxlan-vtep-1)# exit
Leaf1(config)# end

Leaf1# write memory

```

### VTEP Leaf2:

```

Leaf2(config)# interface Loopback 1
Leaf2(conf-if-lol1)# description VTEP
Leaf2(conf-if-lol1)# ip address 10.10.10.2/32

```

```

Leaf2(conf-if-lo1)# exit

Leaf2(config)# ip anycast-address enable
Leaf2(config)# ip anycast-mac-address 00:00:00:00:01:02

Leaf2(config)# interface Vlan 40
Leaf2(conf-if-Vlan40)# ip anycast-address 192.168.40.254/24
Leaf2(conf-if-Vlan40)# neigh-suppress
Leaf2(conf-if-Vlan40)# exit

Leaf2(config)# interface Vlan 41
Leaf2(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf2(conf-if-Vlan41)# neigh-suppress
Leaf2(conf-if-Vlan41)# exit

Leaf2(config)# interface PortChannel 100
Leaf2(conf-if-po100)# switchport trunk allowed vlan add 40,41
Leaf2(conf-if-po100)# exit

Leaf2(config)# interface PortChannel 202
Leaf2(conf-if-po202)# switchport trunk allowed vlan add 40,41
Leaf2(conf-if-po202)# exit

Leaf2(config)# interface vxlan vtep1
Leaf2(conf-if-vxlan-vtep-1)# source-ip 10.10.10.1
Leaf2(conf-if-vxlan-vtep-1)# map vni 400 vlan 40
Leaf2(conf-if-vxlan-vtep-1)# map vni 410 vlan 41
Leaf2(conf-if-vxlan-vtep-1)# exit
Leaf2(config)# end

Leaf2# write memory

```

#### **VTEP Leaf25:**

```

Leaf25(config)# interface Loopback 1
Leaf25(conf-if-lo1)# description VTEP
Leaf25(conf-if-lo1)# ip address 10.10.10.25/32
Leaf25(conf-if-lo1)# exit

Leaf25(config)# ip anycast-address enable
Leaf25(config)# ip anycast-mac-address 00:00:00:00:01:02

Leaf25(config)# interface Vlan 40
Leaf25(conf-if-Vlan40)# ip anycast-address 192.168.40.254/24
Leaf25(conf-if-Vlan40)# neigh-suppress
Leaf25(conf-if-Vlan40)# exit

Leaf25(config)# interface Vlan 41
Leaf25(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf25(conf-if-Vlan41)# neigh-suppress
Leaf25(conf-if-Vlan41)# exit

Leaf25(config)# interface Eth 1/1
Leaf25(conf-if-Eth1/1)# switchport trunk allowed vlan add 40,41
Leaf25(conf-if-Eth1/1)# exit

Leaf25(config)# interface vxlan vtep25
Leaf25(conf-if-vxlan-vtep-25)# source-ip 10.10.10.25
Leaf25(conf-if-vxlan-vtep-25)# map vni 400 vlan 40
Leaf25(conf-if-vxlan-vtep-25)# map vni 410 vlan 41
Leaf25(conf-if-vxlan-vtep-25)# exit
Leaf25(config)# end

Leaf25# write memory

```

#### **VTEP Leaf26:**

```

Leaf26(config)# interface Loopback 1
Leaf26(conf-if-lo1)# description VTEP
Leaf26(conf-if-lo1)# ip address 10.10.10.26/32
Leaf26(conf-if-lo1)# exit

```

```

Leaf26(config)# ip anycast-address enable
Leaf26(config)# ip anycast-mac-address 00:00:00:00:01:02

Leaf26(config)# interface Vlan 40
Leaf26(conf-if-Vlan40)# ip anycast-address 192.168.40.254/24
Leaf26(conf-if-Vlan40)# neigh-suppress
Leaf26(conf-if-Vlan40)# exit

Leaf26(config)# interface Vlan 41
Leaf26(conf-if-Vlan41)# ip anycast-address 192.168.41.254/24
Leaf26(conf-if-Vlan41)#neigh-suppress
Leaf26(conf-if-Vlan41)#exit

Leaf26(config)# interface Eth 1/1
Leaf26(conf-if-Eth1/1)# switchport trunk allowed vlan add 40,41
Leaf26(conf-if-Eth1/1)# exit

Leaf26(config)# interface vxlan vtep26
Leaf26(conf-if-vxlan-vtep-26)# source-ip 10.10.10.26
Leaf26(conf-if-vxlan-vtep-26)# map vni 400 vlan 40
Leaf26(conf-if-vxlan-vtep-26)# map vni 410 vlan 41
Leaf26(conf-if-vxlan-vtep-26)# exit
Leaf26(config)# end

Leaf26# write memory

```

## View asymmetric IRB configuration

### View asymmetric IRB configuration — Multiple tenants

```

Leaf1# show vxlan interface
VTEP Name : vtep-1
VTEP Source IP : 10.10.10.1
EVPN NVO Name : nvo1
EVPN VTEP : vtep-1
Source Interface : Loopback1
PrimaryIP Interface : Not Configured

```

```

Leaf1# show vxlan vlanvnimap
VLAN VNI
===== =====
Vlan41 410
Vlan42 420
Vlan51 510
Vlan52 520
Total count : 4

```

```

Leaf1# show vxlan vrfvnimap
VRF VNI
===== =====
Total count : 0

```

```

Leaf1# show neighbor-suppress-status

VlanId SuppressionStatus

Vlan41 on
Vlan42 on
Vlan51 on
Vlan52 on

```

## VXLAN statistics

To view VXLAN packet statistics in all VXLAN tunnels on a VTEP, use the `show vxlan counters` command. To poll traffic in a specified VXLAN tunnel, enter the destination IP address.

### View VXLAN traffic statistics

```
sonic# show vxlan counters [destination-ip-address]
```

```
sonic# show vxlan counters
Polling Rate : 5 seconds
```

| Interface    | RX_BYTES_OK | RX_OK   | RX_BPS | RX_PPS | TX_BYTES_OK | TX_OK | TX_BPS | TX_PPS |
|--------------|-------------|---------|--------|--------|-------------|-------|--------|--------|
| EVPN_1.1.1.1 | 224452400   | 1122262 | 105349 | 527    | 14670       | 112   | 0      | 0      |
| EVPN_1.1.1.2 | 0           | 0       | 0      | 0      | 0           | 0     | 0      | 0      |

```
sonic# show vxlan counters 1.1.1.1
Polling Rate : 5 seconds
```

| Interface    | RX_BYTES_OK | RX_OK   | RX_BPS | RX_PPS | TX_BYTES_OK | TX_OK | TX_BPS | TX_PPS |
|--------------|-------------|---------|--------|--------|-------------|-------|--------|--------|
| EVPN_1.1.1.1 | 224452400   | 1122262 | 105349 | 527    | 14670       | 112   | 0      | 0      |

- `RX_BYTES_OK` — Total Received bytes over the VXLAN tunnel.
- `RX_OK` — Total VXLAN packets received over the VXLAN tunnel.
- `RX_BPS` — Received bit rate over the VXLAN tunnel.
- `RX_PPS` — Received packet rate over the VXLAN tunnel.
- `TX_BYTES_OK` — Total transmitted bytes over the VXLAN tunnel.
- `TX_OK` — Total VXLAN packets transmitted over the VXLAN tunnel.
- `TX_BPS` — Transmit bit rate over the VXLAN tunnel.
- `TX_PPS` — Transmit packet rate over the VXLAN tunnel.

### Configure polling interval

To configure the polling interval for all VXLAN tunnels on a VTEP, use the `counter polling-interval seconds` command (3 to 30 seconds; default 5).

```
sonic# interface vxlan vxlan-name]
sonic(conf-if-vxlan)# counter polling-interval 10
```

### Clear VXLAN counters

To clear traffic statistics in all VXLAN tunnels on a VTEP, use the `clear counters vxlan [destination-ip-address]` command. To clear traffic statistics for a specified VXLAN tunnel, enter the destination IP address.

```
sonic# clear counters vxlan 1.1.1.1
```

## Configure EVPN

**(i) NOTE:** Ethernet VPN (EVPN) is available only in the Enterprise Standard, Enterprise Premium, and Edge Standard bundles. EVPN is not available in the Cloud Standard and Cloud Premium bundles.

In a VXLAN, EVPN controls the import and export of BGP routes while providing multi-tenant Layer 2/3 VPN services. EVPN functions as a BGP control plane that avoids flood-and-learn to advertise and learn host MAC addresses and MAC/IP bindings in the overlay.

VXLAN carries EVPN routes in external border gateway protocol (eBGP) and internal border gateway protocol (iBGP) sessions in an IP underlay network. Use eBGP or iBGP with EVPN to exchange route information for remote VTEP discovery, and MAC and ARP learning of host devices in L2 tenant segments.

- Each remote VTEP is automatically learned as a member of a VXLAN from the EVPN routes received from the remote VTEP.
- After a remote VTEP address is learned, VXLAN traffic is sent to, and received from, the VTEP.

- Remote host MAC addresses are learned in the control plane using BGP EVPN Type 2 routes and MAC/IP advertisements.
- Symmetric IRB supports Type 5 routes. Type-5 route advertisement is used to communicate between data center networks by advertising the IP prefixes of VXLANs limited to a single data center.

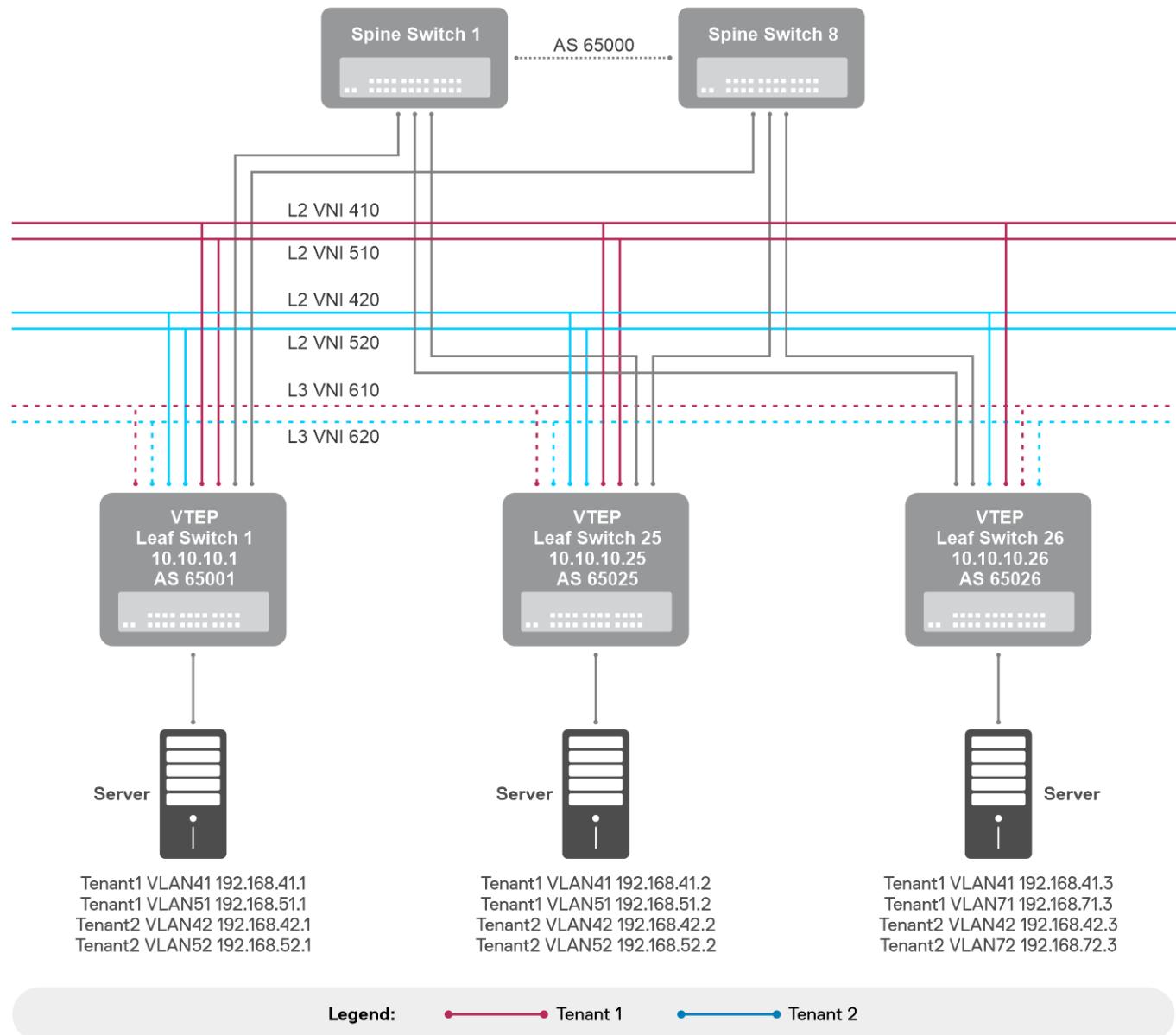
BGP EVPN reduces flooding related to L2 unknown unicast and broadcast traffic. The distribution of host MAC and IP reachability information supports virtual machine (VM) mobility and scalable VXLAN overlay network designs.

After configuring BGP EVPN, a VXLAN starts sending and receiving VXLAN traffic when you configure a VTEP with active VLANs for downstream host devices.

### BGP EVPN example

This sample BGP EVPN topology shows a leaf-spine data center network in which eBGP peer sessions between leaf and spine switches advertise both underlay IP routes and EVPN routes. All spine nodes are in one autonomous system—AS 65000. Each leaf node operates as a VTEP that is associated with a unique autonomous system and connects to downstream servers.

To advertise underlay IP and EVPN routes, eBGP unnumbered peer sessions are established between the leaf and spine nodes using an interface IPv6 link local address.



**Figure 11. BGP EVPN example**

**i** **NOTE:** Although this BGP EVPN example uses spanned tenant VLANs across VTEPs that are configured with different IP addresses, you can use the `ip anycast-address` command to configure the same IP address on all Leaf nodes.

## Before you start

- Configure the eBGP neighbors that serve as VTEPs in the VXLAN.
- For unnumbered BGP in the underlay, enable IPv6 on leaf interfaces with upstream links to spine switches.

## BGP EVPN configuration

To set up BGP EVPN service on each VTEP in a VXLAN overlay network:

- Configure BGP to advertise VTEP source IP addresses to other VTEP neighbors in the underlay network.
- Configure BGP uplinks to spine switches in the underlay network.
- Configure how EVPN routes are advertised and handled in the overlay network.
- Configure each tenant VRF for IPv4/IPv6 route redistribution and L2 EVPN-learned IPv4/IPv6 route advertisement.

### 1. Configure a VTEP to use BGP with EVPN:

- Configure the BGP autonomous system in ROUTER-BGP configuration mode. Assign an IP address to use as the unique BGP router ID.

```
sonic(config)# router bgp as-number
sonic(conf-router-bgp)# router-id ip-address
```

- Configure the exchange of IPv4 or IPv6 unicast routes in the underlay network.

- Configure the redistribution of IPv4 or IPv6 routes that are learned through BGP connections.
- Configure the maximum number of eBGP and iBGP routes that can be selected to form equal cost multi-path (ECMP) routes for load-sharing.

```
sonic(conf-router-bgp)# address-family {ipv4 unicast | ipv6 unicast}
sonic(conf-router-bgp-af)# redistribute connected
sonic(conf-router-bgp-af)# maximum-paths {ibgp number | number}
sonic(conf-router-bgp-af)# exit
sonic(conf-router-bgp) #
```

- Configure a BGP peer group with the configuration settings for uplinks to spine switches:

- Configure a remote autonomous system to exchange routing information through external BGP (eBGP) peer sessions. The `remote-as external` configuration is required for unnumbered BGP.
- (Optional) Configure the time (in seconds) between sending keepalive messages to a BGP neighbor, and the hold-time to wait to receive a keepalive message before considering a BGP peer to be dead.
- (Optional) Configure the route advertisement interval between sending BGP route updates to neighbors.
- (Optional) Enable bi-directional forwarding detection (BFD) to detect forwarding-path failures in BGP routes.
- Enable BGP to negotiate the extended-nexthop capability with a peer. The `capability extended-nexthop` configuration is required for unnumbered BGP.
- Enable the exchange of IPv4 or IPv6 unicast routes with peer-group neighbors.
- Enable the exchange of L2 EVPN routes with peer-group neighbors.

```
sonic(conf-router-bgp) # peer-group peer-group-name
sonic(conf-router-bgp-pg) # remote-as external
sonic(conf-router-bgp-pg) # timers keepalive holdtime
sonic(conf-router-bgp-pg) # advertisement-interval seconds
sonic(conf-router-bgp-pg) # bfd
sonic(conf-router-bgp-pg) # capability extended-nexthop

sonic(conf-router-bgp-pg) # address-family {ipv4 unicast | ipv6 unicast}
sonic(conf-router-bgp-pg-af) # activate
sonic(conf-router-bgp-pg-af) # exit

sonic(conf-router-bgp-pg) # address-family l2vpn evpn
sonic(conf-router-bgp-pg-af) # activate
sonic(conf-router-bgp-pg-af) # exit
sonic(conf-router-bgp-pg) # exit
sonic(conf-router-bgp) #
```

- Configure BGP neighbors by specifying the local interface and assigning the peer-group settings.

```
sonic(conf-router-bgp) # neighbor interface Ethslot/port[/breakout-port]
sonic(conf-router-bgp-neighbor) # description Spine 8
sonic(conf-router-bgp-neighbor) # peer-group peer-group-name
sonic(conf-router-bgp-neighbor) # exit

sonic(conf-router-bgp) # neighbor interface Ethslot/port[/breakout-port]
```

```
sonic(conf-router-bgp-neighbor)# description Spine 1
sonic(conf-router-bgp-neighbor)# peer-group peer-group-name
sonic(conf-router-bgp-neighbor)# exit
sonic(conf-router-bgp) #
```

4. Enable L2VPN EVPN route advertisements in the overlay network:

- Advertise all VNIs configured on the switch. In VXLAN encapsulated frames sent in an overlay network, a VNI identifies an L2 tenant segment.
- (Optional) Advertise all default gateway routes in EVPN, including host MAC/IP bindings.
- (Optional) If the switch is used in an MLAG/logical-VTEP and if you configured a local VXLAN primary IP address, specify the MLAG peer's PIP address. Configure the peer PIP address on each switch in an MLAG when you enable L2VPN EVPN route advertisements.
- (Optional) Filter the IPv4 and IPv6 routes that are advertised by EVPN using a route map. To configure a route map, see [Create a route-map](#).

```
sonic(config)# router bgp as-number [vrf vrf-name]
sonic(conf-router-bgp)# address-family l2vpn evpn
sonic(conf-router-bgp-af)# advertise-all-vni
sonic(conf-router-bgp-af)# advertise-default-gw
sonic(conf-router-bgp-af)# advertise-pip peer-ip peer-pip-address
sonic(conf-router-bgp-af)# advertise {ipv4 unicast | ipv6 unicast} route-map map-name
```

5. Configure any of the following optional settings which determine how L2VPN EVPN routes are handled in the overlay network:

- a. (Optional) Enable the autogeneration of route-target import and export values as described in RFC 8365. A route target (RT) controls the way that EVPN routes are distributed and learned. A receiving VTEP downloads BGP EVPN route information for matching import RT values.

```
sonic(conf-router-bgp-af)# autort rfc8365-compatible
```

- b. (Optional) Enable a border leaf node to originate IPv4 default type-5 EVPN routes.

```
sonic(conf-router-bgp-af)# default-originate {ipv4 | ipv6}
```

- c. (Optional) Configure how a VXLAN switch handles the detection of duplicate MAC addresses of downstream hosts.

- `freeze` — Disables the sending and receiving of route updates for a duplicate MAC address permanently or for a specified time; 30 to 3600 seconds, default 180.
- `max-moves` — Sets the maximum number of local MAC address moves allowed before disabling route advertisement of the address during a specified time; 2 to 1800 seconds, default 180.

```
sonic(conf-router-bgp-af)# dup-addr-detection {freeze {permanent | time seconds} | max-moves number time seconds}
```

- d. (Optional) Manually configure the Route Distinguisher and Route Target values that are used in advertised EVPN routes, where:

- `rd A.B.C.D:[1-65535]` configures the RD with a 4-octet IPv4 address, and an optional 2-byte value.
- `rd auto` automatically generates the RD.
- `route-target {import | export | both} {auto | value}` configures an automatically generated import or export value for EVPN routes or configures a value in the format `2-octet-ASN:4-octet-number` or `4-octet-ASN:2-octet-number`.
  - The 2-octet ASN number is 1 to 65535.
  - The 4-octet ASN number is 1 to 4294967295.
  - To configure the same value for the RT import and export values, use the `both` option.
  - For a L2 VNIs, configure the RD and RT values for each VNI in the BGP L2VPN EVPN address family in the default BGP instance.
  - For a L3 VNIs, configure the RD and RT values for each VNI in the BGP VRF instance.

For a L2 VNI:

```
sonic(conf)# router bgp as-number
sonic(conf-router-bgp)# address-family l2vpn evpn
sonic(conf-router-bgp-af)# vni [1-16777215]
sonic(conf-router-bgp-af-vni)# rd {A.B.C.D:[1-65535] | auto}
sonic(conf-router-bgp-af-vni)# route-target {import | export | both} {auto | value}
```

For a L3 VNI:

```
sonic(conf)# router bgp as-number vrf vrf-name
sonic(conf-router-bgp)# address-family 12vpn evpn
sonic(conf-router-bgp-af)# rd {A.B.C.D:[1-65535] | auto}
sonic(conf-router-bgp-af)# route-target {import | export | both} {auto | value}
```

6. (Optional) If you are using symmetric IRB in your VXLAN network, configure each tenant VRF:

- Configure the redistribution of IPv4 or IPv6 routes.
- Enable the advertisement of IPv4 unicast or IPv6 routes to BGP EVPN as Type-5 routes.

```
sonic(config)# router bgp as-number vrf vrf-name
sonic(conf-router-bgp)# address-family {ipv4 unicast | ipv6 unicast}
sonic(conf-router-bgp-af)# redistribute connected
sonic(conf-router-bgp-af)# exit
sonic(conf-router-bgp)# address-family 12vpn evpn
sonic(conf-router-bgp-af)# advertise {ipv4 unicast | ipv6 unicast}
sonic(conf-router-bgp-af)# end
```

#### Example: BGP EVPN configuration

```
sonic(config)# router bgp 65001
sonic(conf-router-bgp)# router-id 10.0.2.1

sonic(conf-router-bgp)# peer-group SPINE
sonic(conf-router-bgp-pg)# remote-as external
sonic(conf-router-bgp-pg)# timers 3 9
sonic(conf-router-bgp-pg)# advertisement-interval 5
sonic(conf-router-bgp-pg)# bfd
sonic(conf-router-bgp-pg)# capability extended-nexthop
sonic(conf-router-bgp-pg)# address-family ipv4 unicast
sonic(conf-router-bgp-pg-af)# activate
sonic(conf-router-bgp-pg-af)# exit
sonic(conf-router-bgp-pg)# address-family 12vpn evpn
sonic(conf-router-bgp-pg-af)# activate
sonic(conf-router-bgp-pg-af)# exit
sonic(conf-router-bgp-pg)# exit

sonic(conf-router-bgp)# neighbor interface Eth 1/53
sonic(conf-router-bgp-neighbor)# description Spine1
sonic(conf-router-bgp-neighbor)# peer-group SPINE
sonic(conf-router-bgp-neighbor)# exit

sonic(conf-router-bgp)# neighbor interface Eth 1/56
sonic(conf-router-bgp-neighbor)# description Spine8
sonic(conf-router-bgp-neighbor)# peer-group SPINE
sonic(conf-router-bgp-neighbor)# exit

sonic(conf-router-bgp)# address-family ipv4 unicast
sonic(conf-router-bgp-af)# redistribute connected
sonic(conf-router-bgp-af)# maximum-paths 2
sonic(conf-router-bgp-af)# exit

sonic(conf-router-bgp)# address-family 12vpn evpn
sonic(conf-router-bgp-af)# advertise-all-vni
sonic(conf-router-bgp-af)# exit
sonic(conf-router-bgp)# exit
sonic(config) #
```

If you use symmetric IRB and have tenant VRFs configured on the switch, add these commands for each VRF instance:

```
sonic(config)# router bgp 65001 vrf vrf-name
sonic(conf-router-bgp)# address-family ipv4 unicast
sonic(conf-router-bgp-af)# redistribute connected
sonic(conf-router-bgp-af)# exit

sonic(conf-router-bgp)# address-family 12vpn evpn
sonic(conf-router-bgp-af)# advertise ipv4 unicast
sonic(conf-router-bgp-af)# exit
sonic(conf-router-bgp)# exit
```

## View BGP EVPN configuration (symmetric IRB)

```
Leaf1# show evpn
L2 VNIs: 100
L3 VNIs: 10
Advertise gateway mac-ip: Yes
Advertise svi mac-ip: No
Duplicate address detection: Enable
 Detection max-moves 5, time 180
IPv4 Neigh Kernel threshold: 48000
IPv6 Neigh Kernel threshold: 48000
Total IPv4 neighbors: 20244
Total IPv6 neighbors: 6369
```

```
Leaf1# show evpn vni 410
VNI: 410
Type: L2
Tenant VRF: Vrf1
Client State: Up
VxLAN interface: vtep-1-41
VxLAN ifIndex: 77
Local VTEP IP: 10.10.10.1
Local external VTEP IP: 0.0.0.0
VxLAN external interface: unknown
Mcast group: 0.0.0.0
Remote VTEPs for this VNI:
 10.10.10.25 flood: HER
 External: 0, Label: 410
 Kernel Add: Success, Add ReAttempt:0
 10.10.10.26 flood: HER
 External: 0, Label: 410
 Kernel Add: Success, Add ReAttempt:0
Number of MACs (local and remote) known for this VNI: 6
Number of ARPs (IPv4 and IPv6, local and remote) known for this VNI: 4
Advertise-gw-macip: No
```

```
Leaf1# show evpn vni 610
VNI: 610
Type: L3
Tenant VRF: Vrf1
Local Vtep Ip: 10.10.10.1
Local External Vtep Ip: 0.0.0.0
Vxlan-Intf: vtep-1-61
SVI-If: Vlan61
State: Up
Client State: Up
VNI Filter: none
System MAC: 8c:04:ba:a7:eb:c0
Router MAC: 8c:04:ba:a7:eb:c0
L2 VNIs: 410 510
```

## After BGP EVPN configuration: View VXLAN asymmetric IRB operation

```
Leaf1# show vxlan tunnel
Name SIP DIP source Group D-VNI operstatus
===== ====== ====== ====== ====== ====== =====
EVPN_10.10.10.25 10.10.10.1 10.10.10.25 EVPN internal no oper_up
EVPN_10.10.10.26 10.10.10.1 10.10.10.26 EVPN internal no oper_up
```

```
Leaf1# show vxlan remote mac
Vlan Mac Type Tunnel Group VNI
===== ====== ====== ====== ====== =====
Vlan41 00:00:23:32:46:6f dynamic 10.10.10.25 internal 410
Vlan41 00:00:23:32:46:70 dynamic 10.10.10.26 internal 410
Vlan42 00:00:2b:94:e6:05 dynamic 10.10.10.25 internal 420
Vlan42 00:00:2b:94:e6:09 dynamic 10.10.10.26 internal 420
Vlan51 00:00:2b:94:e6:06 dynamic 10.10.10.25 internal 510
Vlan51 00:00:2b:94:e6:0a dynamic 10.10.10.26 internal 510
Vlan52 00:00:2b:94:e6:07 dynamic 10.10.10.25 internal 520
Vlan52 00:00:2b:94:e6:0b dynamic 10.10.10.26 internal 520
```

```
Total count : 8
```

```
Leaf1# show vxlan remote mac 10.10.10.25
Vlan Mac Type Tunnel Group VNI
===== ====== ===== ===== ===== ===
Vlan41 00:00:23:32:46:6f dynamic 10.10.10.25 internal 410
Vlan41 00:00:2b:94:e6:04 dynamic 10.10.10.25 internal 410
Vlan42 00:00:56:90:37:34 dynamic 10.10.10.25 internal 420
Vlan42 00:00:2b:94:e6:05 dynamic 10.10.10.25 internal 420
Vlan51 00:00:2b:94:e6:06 dynamic 10.10.10.25 internal 510
Vlan52 00:00:2b:94:e6:07 dynamic 10.10.10.25 internal 520
Total count : 6
```

```
Leaf1# show vxlan remote vni
Vlan Tunnel Group VNI
===== ====== ===== ===
Vlan41 10.10.10.25 internal 410
Vlan41 10.10.10.26 internal 410
Vlan42 10.10.10.25 internal 420
Vlan42 10.10.10.26 internal 420
Vlan51 10.10.10.25 internal 510
Vlan51 10.10.10.26 internal 510
Vlan52 10.10.10.25 internal 520
Vlan52 10.10.10.26 internal 520
Total count : 8
```

#### After BGP EVPN configuration: View VXLAN symmetric IRB operation

```
Leaf1# show evpn rmac vni 610
Number of Remote RMACs known for this VNI: 2
MAC Remote VTEP
00:00:10:10:25:01 10.10.10.25
00:00:10:10:26:01 10.10.10.26
```

#### View Type-2 routes

```
Leaf1# show bgp l2vpn evpn route type macip
BGP table version is 915, local router ID is 10.0.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]
 Network Next Hop Metric LocPrf Weight Path
 Extended Community
Route Distinguisher: 10.0.2.1:41
*> [2]:[0]:[48]:[00:50:56:6f:ad:a7]
 10.10.10.1 32768 i
 ET:8 RT:65001:410
*> [2]:[0]:[48]:[00:50:56:90:22:bb]
 10.10.10.1 32768 i
 ET:8 RT:65001:410
*> [2]:[0]:[48]:[00:50:56:90:6d:f1]
 10.10.10.1 32768 i
 ET:8 RT:65001:410
Route Distinguisher: 10.0.2.1:51
*> [2]:[0]:[48]:[00:50:56:90:48:ee]
 10.10.10.1 32768 i
 ET:8 RT:65001:510
Route Distinguisher: 10.0.2.25:41
*> [2]:[0]:[48]:[00:00:2b:94:e6:04]
 10.10.10.25 0 65000 65025 i
 RT:65025:410 ET:8 MM:1
* [2]:[0]:[48]:[00:00:2b:94:e6:04]
 10.10.10.25 0 65000 65025 i
 RT:65025:410 ET:8 MM:1
*> [2]:[0]:[48]:[00:00:2b:94:e6:04]:[32]:[192.168.41.202]
 10.10.10.25 0 65000 65025 i
 RT:65025:410 ET:8 MM:1
```

```

* [2]:[0]:[48]:[00:00:2b:94:e6:04]:[32]:[192.168.41.202]
 10.10.10.25 0 65000 65025 i
 RT:65025:410 ET:8 MM:1
*> [2]:[0]:[48]:[00:50:56:90:37:34]
 10.10.10.25 0 65000 65025 i
 RT:65025:410 ET:8
* [2]:[0]:[48]:[00:50:56:90:37:34]
 10.10.10.25 0 65000 65025 i
 RT:65025:410 ET:8
...
Displayed 22 prefixes (40 paths) (of requested type)

```

### View Type-5 routes

```

Leaf1# show bgp l2vpn evpn route type prefix
BGP table version is 2, local router ID is 10.0.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]
 Network Next Hop Metric LocPrf Weight Path
 Extended Community
Route Distinguisher: 192.168.51.254:5096
*> [5]:[0]:[24]:[192.168.41.0]
 10.10.10.1 0 32768 ?
 ET:8 RT:65001:610 Rmac:8c:04:ba:a7:eb:c0
*> [5]:[0]:[24]:[192.168.51.0]
 10.10.10.1 0 32768 ?
 ET:8 RT:65001:610 Rmac:8c:04:ba:a7:eb:c0
Route Distinguisher: 192.168.52.254:5097
*> [5]:[0]:[24]:[192.168.42.0]
 10.10.10.1 0 32768 ?
 ET:8 RT:65001:620 Rmac:8c:04:ba:a7:eb:c0
*> [5]:[0]:[24]:[192.168.52.0]
 10.10.10.1 0 32768 ?
 ET:8 RT:65001:620 Rmac:8c:04:ba:a7:eb:c0
Route Distinguisher: 192.168.71.254:5096
* [5]:[0]:[24]:[192.168.41.0]
 10.10.10.26 0 65000 65026 ?
 RT:65026:610 ET:8 Rmac:8c:04:ba:a7:ee:c0
*> [5]:[0]:[24]:[192.168.41.0]
 10.10.10.26 0 65000 65026 ?
 RT:65026:610 ET:8 Rmac:8c:04:ba:a7:ee:c0
* [5]:[0]:[24]:[192.168.71.0]
 10.10.10.26 0 65000 65026 ?
 RT:65026:610 ET:8 Rmac:8c:04:ba:a7:ee:c0
*> [5]:[0]:[24]:[192.168.71.0]
 10.10.10.26 0 65000 65026 ?
 RT:65026:610 ET:8 Rmac:8c:04:ba:a7:ee:c0
Route Distinguisher: 192.168.72.254:5097
*> [5]:[0]:[24]:[192.168.42.0]
 10.10.10.26 0 65000 65026 ?
 RT:65026:620 ET:8 Rmac:8c:04:ba:a7:ee:c0
* [5]:[0]:[24]:[192.168.42.0]
 10.10.10.26 0 65000 65026 ?
 RT:65026:620 ET:8 Rmac:8c:04:ba:a7:ee:c0
*> [5]:[0]:[24]:[192.168.72.0]
 10.10.10.26 0 65000 65026 ?
 RT:65026:620 ET:8 Rmac:8c:04:ba:a7:ee:c0
* [5]:[0]:[24]:[192.168.72.0]
 10.10.10.26 0 65000 65026 ?
 RT:65026:620 ET:8 Rmac:8c:04:ba:a7:ee:c0
Displayed 8 prefixes (12 paths) (of requested type)

```

## Filter Type-2, Type-3, and Type-5 EVPN routes

To quickly find Type-2, Type-3, and Type-5 EVPN routes in show outputs, you can filter the `show bgp l2vpn evpn route type {macip | prefix | multicast} display`.

- Filter Type-5 prefix routes by specifying an IPv4/IPv6 subnet and mask.

```
sonic# show bgp l2vpn evpn route type prefix ip ip-address/mask

sonic# show bgp l2vpn evpn route type prefix ip 10.116.4.87/32

BGP table version is 1, local router ID is 10.116.4.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]:[IPlen]:[VTEP-IP]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]
 Network Next Hop Metric LocPrf Weight Path
 Extended Community
Route Distinguisher: 10.116.4.25:5096
* i [5]:[0]:[32]:[10.116.4.87]
 10.116.4.167 0 100 0 64555 ?
 RT:64513:4060 ET:8 Rmac:04:f8:f8:6b:0f:94
*> i [5]:[0]:[32]:[10.116.4.87]
 10.116.4.167 0 100 0 64555 ?
 RT:64513:4060 ET:8 Rmac:04:f8:f8:6b:0f:94
Route Distinguisher: 10.116.4.26:5096
* i [5]:[0]:[32]:[10.116.4.87]
 10.116.4.167 0 100 0 64555 ?
 RT:64513:4060 ET:8 Rmac:04:f8:f8:6b:0f:94
*> i [5]:[0]:[32]:[10.116.4.87]
 10.116.4.167 0 100 0 64555 ?
 RT:64513:4060 ET:8 Rmac:04:f8:f8:6b:0f:94
Route Distinguisher: 10.116.4.65:5096
*> [5]:[0]:[32]:[10.116.4.87]
 10.116.4.200 32768 64555 i
 ET:8 RT:64513:4060 Rmac:04:f8:f8:6b:3b:94
Displayed 3 prefixes (5 paths) (of requested type)
```

- Filter Type-2 MAC routes by specifying a MAC address.

```
sonic# show bgp l2vpn evpn route type macip mac mac-address

sonic# show bgp l2vpn evpn route type macip mac 04:f8:f8:6b:39:94

BGP table version is 2, local router ID is 10.116.4.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]:[IPlen]:[VTEP-IP]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]
 Network Next Hop Metric LocPrf Weight Path
 Extended Community
Route Distinguisher: 10.116.4.25:300
* i [2]:[0]:[48]:[04:f8:f8:6b:39:94]
 10.116.4.167 0 100 0 i
 RT:64513:300 ET:8
*> i [2]:[0]:[48]:[04:f8:f8:6b:39:94]
 10.116.4.167 0 100 0 i
 RT:64513:300 ET:8
Displayed 1 prefixes (2 paths) (of requested type)
```

- Filter Type-2 MAC/IP routes by specifying a MAC address and IPv4/IPv6 host address without a mask.

```
sonic# show bgp 12vpn evpn route type macip mac mac-address ip ip-address

sonic# show bgp 12vpn evpn route type macip mac 04:f8:f8:7e:0f:37 ip 38.2.0.4

BGP table version is 26, local router ID is 10.116.4.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]:[IPlen]:[VTEP-IP]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]
Network Next Hop Metric LocPrf Weight Path
 Extended Community
Route Distinguisher: 10.116.4.65:3802
*> [2]:[0]:[48]:[04:f8:f8:7e:0f:37]:[32]:[38.2.0.4]
 10.116.4.200 32768 i
 ET:8 RT:64513:3802 RT:64513:3801 Rmac:04:f8:f8:6b:3b:94
Displayed 1 prefixes (1 paths) (of requested type)
```

- Filter Type-3 IMET routes by specifying a VTEP source-ip IPv4 address. To discover remote peers and set up tunnels for BUM traffic over VXLAN, BGP EVPN uses Type-3 routes — also known as Inclusive Multicast Ethernet Tag (IMET) routing.

```
sonic# show bgp 12vpn evpn route type multicast ip source-ip-address
```

```
sonic# show bgp 12vpn evpn route type multicast ip 10.116.5.210

BGP table version is 1, local router ID is 10.116.4.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]:[IPlen]:[VTEP-IP]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]
Network Next Hop Metric LocPrf Weight Path
 Extended Community
Route Distinguisher: 10.116.5.65:200
*> [3]:[0]:[32]:[10.116.5.210]
 10.116.5.210 0 64514 i
 RT:64514:200 ET:8
Displayed 1 prefixes (1 paths) (of requested type)
```

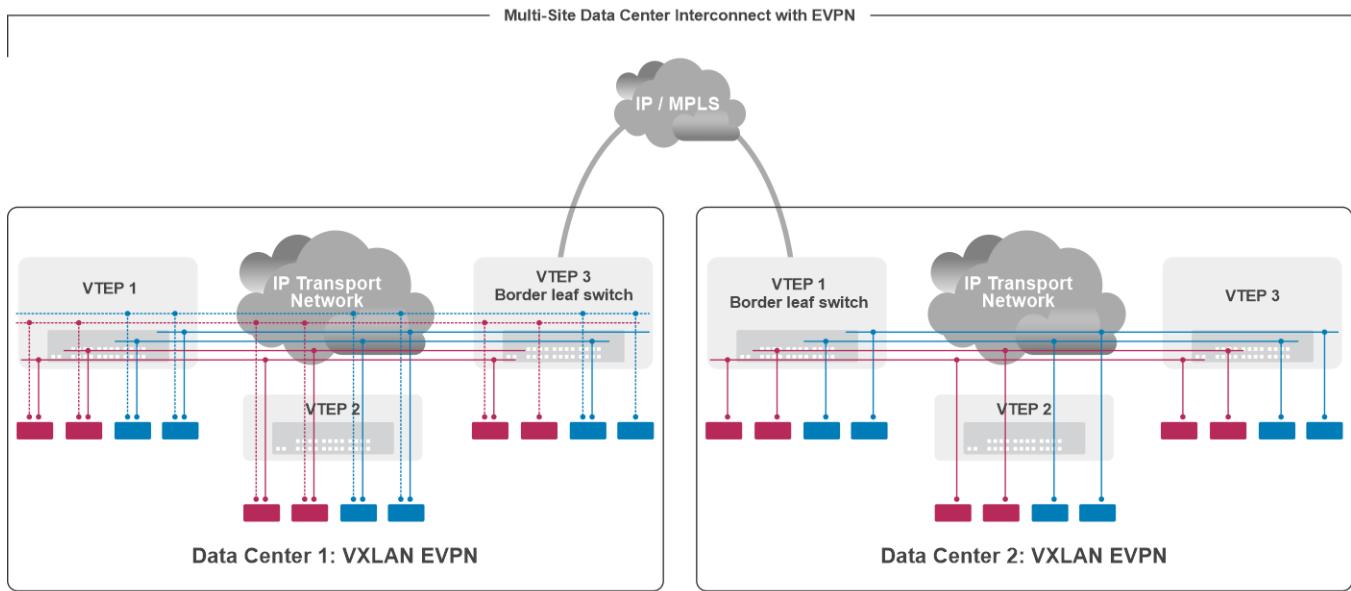
## Multi-site data center interconnect

**(i) NOTE:** Multi-site DCI is available only in the Enterprise Standard and Enterprise Premium bundles. It is not available in the Cloud Standard, Cloud Premium, and Edge Standard bundles.

To interconnect two or more VXLAN EVPN networks over an IP network, use the multi-site data center interconnect (DCI) feature. Multi-site DCI uses border leaf VTEPs to terminate VXLAN tunnels from, and originate VXLAN tunnels to remote sites.

Each border VTEP imports and re-originates BGP updates, and decapsulates and re-encapsulates VXLAN data traffic crossing it. A border VTEP terminates and re-originates routes, and the data traffic to external sites and to VTEPs in the local site. Each intrasite VTEP supports Type-2, Type 3, and Type 5 routes in the VXLAN EVPN overlay control plane.

Enabling and disabling multi-site DCI on a border VTEP does not impact the existing VXLAN EVPN configuration on the VTEPs in a site.



**Figure 12. Multi-site Data Center Interconnect**

On a border VTEP, BGP performs the following tasks:

- Re-advertises routes from internal to external neighbors (and vice versa) using itself as the next-hop.
- Re-advertises only those routes which are imported on the border VTEP.
- Does not allow routes which have a local scope, such as Type-1 and Type-4, to be propagated between internal and external neighbors.
- Does not allow Type-3 routes to be propagated between internal and external neighbors. A border VTEP advertises its local Type-3 routes to internal and external VTEPs.

### Multi-site DCI configuration

To set up multi-site DCI, configure VXLAN and EVPN as described in [Configure VXLAN](#) and [Configure EVPN](#). Then on each border leaf VTEP, follow these steps:

1. To distinguish between internal and external VXLAN tunnels, configure a separate source and external IP address on each border leaf VTEP. In MLAG, configure the same source and external IP address for remote site connections on each MLAG peer. A source IP is required before you create VLAN-VNI mappings; for example:

```
sonic(config) # interface vxlan vtep-1
sonic(conf-if-vxlan-vtep-1) # source-ip 192.168.10.1
sonic(conf-if-vxlan-vtep-1) # external-ip 10.1.1.1
```

2. Configure the BGP EVPN neighbors in remote sites as external fabric neighbors. To identify a BGP peer in a remote data center site, configure the neighbor as `fabric-external` in the L2VPN EVPN address family:

```
sonic(config) # router bgp local-asn
sonic(config-router-bgp) # neighbor remote-ip-address
sonic(config-router-bgp-neighbor) # remote-as external
sonic(config-router-bgp-neighbor) # address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af) # activate
sonic(config-router-bgp-neighbor-af) # fabric-external
```

On a border leaf, IP routes are handled as follows:

- Type-2 and Type-5 routes are received and advertised between internal and external neighbors. These routes are re-originated with the local route-distinguisher (RD) and route-targets (RT) configured for the MAC/IP-VRF instance. The original route-targets in the route are stripped in the re-originated route. If the route is received from an external BGP neighbor, the re-originated route is advertised only to internal neighbors, and vice versa. The Type-2 routes that are converted to (/32 or /128) host routes are not re-originated as Type-5 routes.
- Type-3 (IMET) routes received from external and internal neighbors are not re-originated. Instead, these routes are consumed locally. Only the local IMET routes are advertised to external and internal neighbors.

- Type-1 and Type-4 routes are not advertised by a border leaf. If these routes are received from an internal or fabric-external neighbor, they are locally consumed and not re-originated. To configure custom control on routes crossing a border leaf, apply inbound and outbound policies on external/internal neighbors to use special communities or extended communities in route updates.
3. (Optional) In a data center network, VTEPs usually have the same VLAN-VNI mapping. However, the remote sites to which a border leaf connects may be in a separate administrative domain and have different VNI assignments. In this case, for VTEPs in an external fabric, you can configure support for VNI assignment to be downstream in the local fabric. Enter `vni downstream external` to enable support for downstream VNI assignment for all external VTEPs.

```
sonic(config)# interface vxlan vtep-1
sonic(conf-if-vxlan-vtep-1)# vni downstream external
```

If there are only specific internal or external VTEPs that use different VLAN-VNI assignments, enable the downstream VNI assignment for specified remote VTEP IP addresses; for example:

```
sonic(config)# interface vxlan vtep-1
sonic(conf-if-vxlan-vtep-1)# vni downstream 10.10.10.10
sonic(conf-if-vxlan-vtep-1)# vni downstream 10.10.10.11
```

**i | NOTE:** If you have already configured `vni downstream external`, it is not necessary to use `vni downstream` to enable downstream VNI assignment for specified remote VTEP IP addresses.

4. (Optional) When you enable downstream VNI, configure the route targets for L2 and L3 VNIs in the BGP L2VPN EVPN address family. Enable auto-RT and manual RT at the same time: border leafs with multi-site DCI import routes from internal VTEPs using auto-RT, and import routes from external VTEPs using manually specified RT values. For example, to manually configure the import/export of routes in a specified VNI or VRF using auto-generated RT values:

```
! VNI configuration
sonic(config)# router bgp 10
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# vni 1010
sonic(config-router-bgp-af-vni)# route-target both 1:1010
sonic(config-router-bgp-af-vni)# route-target {import|export|both} auto
sonic(config-router-bgp-af-vni)# end
! BGP VRF L2VPN EVPN configuration
sonic(config)# router bgp 10 vrf Vrf-red
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# route-target both 1:10010
sonic(config-router-bgp-af)# route-target {import|export|both} auto
```

**i | NOTE:** If you do not manually configure RTs, auto-RT is enabled by default to import and export routes. When you manually configure RT values, you must also configure the import/export of routes using auto-RT. You may use auto-RT for both local and remote sites. Then there is no need to configure manual RTs or auto-RT.

To import routes in a VNI or VRF by ignoring the Global Administrator field in the RTs of incoming routes:

```
! VNI configuration
sonic(config-router-bgp-af)# vni 1010
sonic(config-router-bgp-af-vni)# route-target import *:value
sonic(config-router-bgp-af-vni)# exit
! BGP VRF L2VPN EVPN configuration
sonic(config-router-bgp-af)# route-target import *:value
```

5. (Optional) If a border leaf is an MLAG peer, configure a unique primary IP address (PIP) on each border MLAG node that is advertised to internal and external VTEPs. The primary IP is used as the next-hop for the local routes advertised by an MLAG border leaf instead of using the logical VTEP IP address of the MLAG as the source. EVPN L3VNI (Type-5) and Type-2 routes for local orphan hosts are advertised to internal and external BGP peers with the primary IP as the next-hop. For example, on each MLAG peer, configure:

```
sonic(config)# interface vxlan vxlan-interface-name
sonic(conf-if-vtep1)# primary-ip 2.2.2.2
sonic(conf-if-vtep1)# exit

sonic(config)# interface Loopback 2
sonic(conf-if-lo2)# ip address 2.2.2.2/32
sonic(conf-if-lo2)# exit
```

```
sonic(config)# router bgp 10 vrf default
sonic(conf-router)# router-id 2.2.2.2
```

6. (Optional) Enable neighbor suppression in a host VLAN to avoid unnecessary flooding of broadcast ARP and Neighbor Discovery requests from local edge ports.

```
sonic(config)# interface vlan vlan-id
sonic(conf-if-vlan)# neigh-suppress
```

#### Example: Multi-site DCI configuration

```
! Configure MLAG Node 1
sonic(config)# interface vxlan vtep-1
sonic(conf-if-vxlan-vtep-1)# source-ip 192.168.1.1
sonic(conf-if-vxlan-vtep-1)# external-ip 10.10.10.10
sonic(conf-if-vxlan-vtep-1)# primary-ip 192.168.1.2
sonic(conf-if-vxlan-vtep-1)# vni downstream external
sonic(conf-if-vxlan-vtep-1)# exit

! Configure local VLAN-VNI assignments here
sonic(config)# interface vxlan vtep-1
sonic(conf-if-vtep)# map vni 410 Vlan 41
sonic(conf-if-vtep)# map vni 410 Vrf1
sonic(conf-if-vtep)# exit

! Configure BGP
sonic(config)# router bgp 10
sonic(config-router-bgp)# router-id 192.168.1.2
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-af)# advertise-all-vni
sonic(config-router-af)# advertise-pip peer-ip 192.168.1.3
sonic(config-router)# neighbor 192.168.10.1 ! Internal peer
sonic(config-router-bgp-neighbor)# remote-as external
sonic(config-router-bgp-neighbor-af)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router)# neighbor 10.1.10.1
sonic(config-router-bgp-neighbor)# remote-as external
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# fabric-external ! External peer

! Configure MLAG Node 2
sonic(config)# interface vxlan vtep-2
sonic(conf-if-vxlan-vtep-2)# source-ip 192.168.1.1
sonic(conf-if-vxlan-vtep-2)# external-ip 10.10.10.10
sonic(conf-if-vxlan-vtep-2)# primary-ip 192.168.1.3
sonic(conf-if-vxlan-vtep-2)# vni downstream external

! Configure local VLAN-VNI assignments
sonic(config)# interface vxlan vtep-2
sonic(conf-if-vtep)# map vni 410 Vlan 41
sonic(conf-if-vtep)# map vni 410 Vrf1
sonic(conf-if-vtep)# exit

! Configure BGP
sonic(config)# router bgp 10
sonic(config-router-bgp)# router-id 192.168.1.3
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise-all-vni
sonic(config-router-bgp-af)# advertise-pip peer-ip 192.168.1.2

! Add BGP neighbors here as configured on node1
sonic(config-router)# neighbor 192.168.20.2
sonic(config-router-bgp-neighbor)# remote-as external
sonic(config-router-bgp-neighbor-af)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router)# neighbor 10.1.20.2
sonic(config-router-bgp-neighbor)# remote-as external
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# fabric-external
```

## View Multi-site DCI configuration

To check whether a BGP EVPN neighbor is configured as a neighbor in an external fabric:

```
sonic# show bgp ipv4 unicast neighbors

BGP neighbor on Ethernet0: fe80::5054:ff:fece:4c39, remote AS 10, local AS 20, external
link
Hostname: sonic
 BGP version 4, remote router ID 100.0.0.1, local router ID 2.2.2.2
 BGP state = Established, up for 00:00:45
...
For address family: IPv4 Unicast
 Update group 3, subgroup 3
 Packet Queue length 0
 Community attribute sent to this neighbor(all)
 0 accepted prefixes

For address family: L2VPN EVPN
 Update group 4, subgroup 4
 Packet Queue length 0
Fabric-external
 NEXT_HOP is propagated unchanged to this neighbor
 Community attribute sent to this neighbor(all)
 1 accepted prefixes

 Connections established 3; dropped 2
...
```

To view the IP address of an external VTEP:

```
sonic# show vxlan interface

VTEP Name : vtep1
VTEP Source IP : 1.1.1.1
VTEP Primary IP : 2.2.2.2
VTEP External IP : 10.10.10.10
EVPN NVO Name : nvo1
EVPN VTEP : vtep1
Source Interface : Loopback10
Primary IP interface : Loopback20
External IP interface: Loopback30
```

To view the VXLAN tunnels with an external VTEP, where:

- Group — Indicates an internal or external destination VTEP.
- D-VNI — Displays whether a VNI assignment on a remote VTEP is downstream (yes) or global (no).

```
sonic# show vxlan tunnel

Name SIP DIP source Group D-VNI operstatus
===== ====== ====== ===== ===== ===== =====
EVPN_1.0.1.1 1.0.1.255 1.0.1.1 EVPN external yes oper_down
EVPN_1.0.3.1 1.0.1.255 1.0.3.1 EVPN external yes oper_up
EVPN_1.0.3.255 1.0.1.255 1.0.3.255 EVPN internal no oper_up
EVPN_1.0.4.1 1.0.1.255 1.0.4.1 EVPN external yes oper_up
EVPN_1.0.5.1 1.0.1.255 1.0.5.1 EVPN external yes oper_up
```

On an EVPN VTEP, to view whether the VNI for a host VLAN is reachable through an internal or external VTEP:

```
sonic# show vxlan remote vni
Vlan Tunnel Group VNI
===== ====== ===== ====
Vlan1001 1.0.3.255 internal 1001
Vlan1001 1.0.5.1 external 100001
Vlan1002 1.0.3.255 internal 1002
Vlan1002 1.0.5.1 external 100002
```

On an EVPN VTEP, to view whether the MAC address received for a host VLAN is reachable through an internal or external VTEP:

```
sonic# show vxlan remote mac
Vlan Mac Type Tunnel Group VNI
===== ====== ====== ====== ====== =====
Vlan1001 3c:2c:99:2d:7d:38 dynamic 1.0.3.255 internal 1001
Vlan1001 3c:2c:99:6d:da:4c dynamic 1.0.3.255 internal 1001
Vlan1002 3c:2c:99:2d:7d:38 dynamic 1.0.3.255 internal 1002
Vlan1002 3c:2c:99:6d:da:4c dynamic 1.0.3.255 internal 1002
Vlan1003 00:c0:05:31:00:01 dynamic 1.0.5.1 external 10003
Vlan1003 00:c0:05:31:00:02 dynamic 1.0.5.1 external 10003
...
```

## EVPN multihoming

To provide redundant VXLAN EVPN connectivity for downstream tenant devices, use EVPN multihoming. EVPN multihoming allows a tenant VLAN to connect to more than one VTEP for Layer 2/3 VPN services and importing or exporting EVPN routes. The all-active multihomed VTEPs must be connected to a VXLAN EVPN network using a port channel (LAG) running LACP. The multihomed VTEPs in the port channel create an Ethernet segment. All-active multihoming means that load balancing is performed on the traffic flow to and from a tenant device in the LACP port channel, and that the MAC addresses and MAC/IP bindings learned on one VTEP are stored on all multihomed VTEPs in the Ethernet segment.

EVPN multihoming supports RFCs 7432 and 8365. Although MCLAG is also supported as a redundancy solution to connect downstream tenant devices to a VXLAN network, multihoming and legacy MCLAG functionality are mutually exclusive. EVPN multihoming and MCLAG operation are not supported at the same time on a switch.

### EVPN multihoming usage notes

- You can configure up to four VTEPs together for EVPN-based active-active access redundancy. All multihomed VTEPs must run Enterprise SONiC; other vendor NOSs are not supported.
- EVPN multihoming supports all routing and switching traffic flows that are supported in MCLAG.
- EVPN multihoming supports these features:
  - Static anycast gateway — Required on multihomed VTEPs to achieve an active-active L3 gateway.
  - DHCP relay — DHCP relay is agnostic to an underlying multihoming configuration. The link-select and source-interface configurations are required for correct DHCP Relay operation in multihomed VTEPs.
  - Startup delay
  - Uplink tracking
  - BGP over EVPN multihomed Ethernet segment
  - ARP/Neighbor Discovery (ND) suppression — A multihomed VTEP responds to ARP/ND requests received on local access ports only for ARP/ND operation with remote VTEPs. The VTEP does not generate an ARP/ND response if ARP/ND is configured on the local multihomed Ethernet segment even though ARP/ND learning is performed on a remote VTEP in the segment.
- Single-active redundancy on a multihomed VTEP is not supported; only active-active redundancy is supported.
- An EVPN multihomed interface that is configured as a router port or a routed subinterface is not supported.
- An EVPN multihomed interface must be configured as an L2 switchport.
- VRRP over EVPN multihoming is not recommended. Use a static anycast gateway instead.
- EVPN multihoming does not support the synchronization of STP states between multihomed VTEPs to avoid L2 loops in the network.
- EVPN multihoming is not supported between Border Gateway routers in a multisite data center interconnect (DCI) deployment that interconnects two VXLAN EVPN networks over an IP network.
- IGMP snooping is not supported on EVPN multihomed interfaces.
- An MCLAG peer gateway is not supported. Use outbound route maps to send routes with an anycast IP address.
- Type-0, Type-1, and Type-3 Ethernet segment IDs are supported.
  - A Type-0 ES-ID is user-configured.
  - A Type-1 ES-ID is automatically derived from the LACP peer's MAC address.
  - A Type-3 ES-ID is automatically derived by combining the configured system-mac address on a port-channel interface and the port-channel number.
- Ensure that the configurations on each multihomed VTEP in an Ethernet segment are the same, including the port-channel connection, System MAC address, Ethernet segment type and ID, VLAN IDs, VNIs, tenant VLAN members connected to the Ethernet segment, and static MAC/neighbor configurations.

## Configure EVPN multihoming

1. Configure the EVPN Ethernet segment on each port channel that connects to multihomed VTEPs, including the System MAC address.
  - a. Configure the Ethernet segment ID. Be sure to configure the same ES-ID on each VTEP in an Ethernet segment. Type-0, Type-1, and Type-3 ES-ID types are supported (see [RFC 7432](#)). To configure a Type-0 ES-ID, enter a 10-byte ID with the type byte set to 0 in the format XX:XX:XX:XX:XX:XX:XX:XX:XX; for example:

```
sonic(config)# interface PortChannel1
sonic(config-if-pol1)# system-mac 00:00:00:0a:00:01
sonic(config-if-pol1)# evpn ethernet-segment 00:00:00:00:00:00:00:0a:00:01
sonic(config-es-id-00:00:00:00:00:00:00:0a:00:01) #
```

**i | NOTE:** In addition to the Ethernet segment ID, a System MAC address is always required in an EVPN Ethernet segment configuration.

To specify an automatically generated Type-1 ES-ID, enter the `auto-lACP` option.

```
sonic(config)# interface PortChannel1
sonic(config-if-pol1)# system-mac 00:00:00:0a:00:01
sonic(config-if-pol1)# evpn ethernet-segment auto-lACP
sonic(config-es-id-auto-lACP) #
```

A Type-1 ES-ID has the following format: 1-byte circuit type (0x01), 6-byte MAC address of LACP partner, 2-byte port-channel number, and 1-byte (0x00).

```
0x01mac-address-LACP-partnerportchannel-number0x00
```

To specify an automatically generated Type-3 ES-ID, enter the `auto-system-mac` option. The Type-3 ES-ID contains the port channel's System MAC address and interface number; for example:

```
sonic(config)# interface PortChannel1
sonic(config-if-pol1)# system-mac 00:00:00:0a:00:01
sonic(config-if-pol1)# evpn ethernet-segment auto-system-mac
sonic(config-es-id-auto-system-mac) #
```

**i | NOTE:** A 10-byte Type-3 ES-ID is generated by concatenating the 6-byte System MAC, the 3-byte Ethernet segment number, and the 1-byte Type=0x03.

- b. (Optional) Configure a Designated Forwarder (DF) preference value (1-65535) to determine which VTEP in an active-active Ethernet segment forwards BUM traffic (see [RFC 7432](#)). By default, the VTEP with the highest DF value is selected as the designated forwarder. If the configured DF values are the same, the VTEP with the lowest source IP address is chosen as the DF.

```
sonic(config)# interface PortChannel1
sonic(config-if-pol1)# system-mac 00:00:00:0a:00:01
sonic(config-if-pol1)# evpn ethernet-segment 00:11:22:33:44:55:66:77:88:01
sonic(config-es-id-00:11:22:33:44:55:66:77:88:01) # df-preference 1
```

2. Configure global multihoming settings on each VTEP in an EVPN Ethernet segment.

- a. Configure a startup delay in seconds (0-3600; default 300) to avoid traffic loss during the VTEP bootup process. During VTEP bootup, the EVPN multihoming interfaces are kept in an administrative-down state until the startup-delay timer expires. As a result, traffic from a multihomed tenant device is not load-balanced to the VTEP until the VTEP starts up and is ready.

```
sonic(config)# evpn esi-multihoming
sonic(config-evpn-esi-mh) # [no] startup-delay seconds
```

**i | NOTE:** In order for the startup delay timer to take effect, L3 interface tracking on uplinks must be configured (see [Interface tracking](#)).

- b. Configure the hold time in seconds (0-86400; default 1080) that is used to wait before aging out the MAC addresses of downstream devices that are learned from multihomed peer VTEPs and that have not been used. The default value is recommended. Increase the MAC hold time when you increase the scale of an EVPN Ethernet segment.

```
sonic(config)# evpn esi-multihoming
sonic(config-evpn-esi-mh) # [no] mac-holdtime seconds
```

- i** **NOTE:** When a MAC address is deleted from the multihomed VTEP on which it is learned, the timer is started on the remote multihomed VTEPs in the EVPN Ethernet segment, which continue to advertise the downstream MAC address.
- c. Configure the hold time in seconds (0-86400; default 1080) that is used to wait before aging out ARP/ND entries that are learned from multihomed peer VTEPs and that have not been used.

```
sonic(config)# evpn esi-multihoming
sonic(config-evpn-esi-mh)# [no] neigh-holdtime seconds
```

- i** **NOTE:** When an ARP entry is deleted from the multihomed VTEP on which it is learned, the timer is restarted on the remote multihomed VTEPs in the EVPN Ethernet segment, which continue to advertise the IP-MAC address association.
- d. Configure uplink tracking to avoid black-holing traffic from a tenant device in case an EVPN multihomed VTEP gets isolated from the network. The multihomed downlink interfaces on the VTEP are shut down if all uplink interfaces on the VTEP go down. You can specify an optional timeout value (in seconds) to wait before bringing up a multihomed interface after one or more uplink interfaces come up. Create a track group and associate all EVPN multihoming interfaces to the track group; for example:

```
sonic(config)# link state track trackGrp
sonic(config-link-track)# downstream all-evpn-es
sonic(config-link-track)# timeout 300
sonic(config-link-track)# exit
sonic(config)# interface Ethernet0
sonic(config-if-Ethernet0)# link state track trackGrp upstream
```

- e. To configure multiple VTEPs together with EVPN-based active-active L3 access redundancy, you must manually configure the same static anycast gateway on each multihomed VTEP.

```
sonic(config)# ip anycast mac-address mac-address
```

- i** **NOTE:** In BGP over EVPN multihoming operation, SVI interfaces on multihomed VTEPs must be configured with a unique IP address on each VTEP.
3. On Trident4-based platforms — Z9432F-ON and S5448F-ON — the switch-resource configuration is required to install remote Type-2 (non-zero ESI) EVPN multihomed MAC addresses with Layer 2 next-hop groups (ECMP) in the ASIC. The switch-resource configuration is required even if the Ethernet segment ID is not configured locally or if MLAG is configured.

```
sonic(config)# switch-resource
sonic(config-switch-resource)# 12-nexthop-group
```

- i** **NOTE:** The switch-resource configuration reduces the MAC and ARP scalability by half.
4. (Optional) Enterprise SONiC advertises EAD-per-EVI routes by default to BGP VTEP neighbors. If third-party VTEP switches do not support EAD-per-EVI routes, disable the learning and advertisement of EAD-per-EVI routes on an Enterprise SONiC multihomed switch:

```
sonic(config)# router bgp 10
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# [no] disable-ead-evi-rx
sonic(config-router-bgp-af)# [no] disable-ead-evi-tx
```

### Example: EVPN multihoming configuration

This example shows a typical EVPN configuration with multihomed (MH) and single-homed (SH) tenant devices connected to VXLAN EVPN VTEPs. In the EVPN multihoming example, tenant devices H1, H4 and H5 are single-homed; tenant devices H2 and H3 are multi-homed. An active-active redundancy with four VTEPs is used, although you can connect multihomed devices to a smaller number of VTEPs.

Tenant devices H1, H4, and H5 use single-homed connections to VTEP 1, VTEP 4, and VTEP 5, respectively. Tenant devices H2 and H3 use multihomed connections to the VXLAN EVPN network with active-active LAG redundancy. H2 is multihomed to VTEP 1 and VTEP 4. H3 is multihomed to VTEP 1, VTEP 2, VTEP 3, and VTEP 4.

To configure EVPN multihoming on the port channels that connect multihomed tenant devices:

- You must configure a system MAC address and a Type-0, Type-1, or Type-3 Ethernet segment ID (ES-ID) on each multihomed port-channel interface. There is no advantage to using one ES-ID type over another.

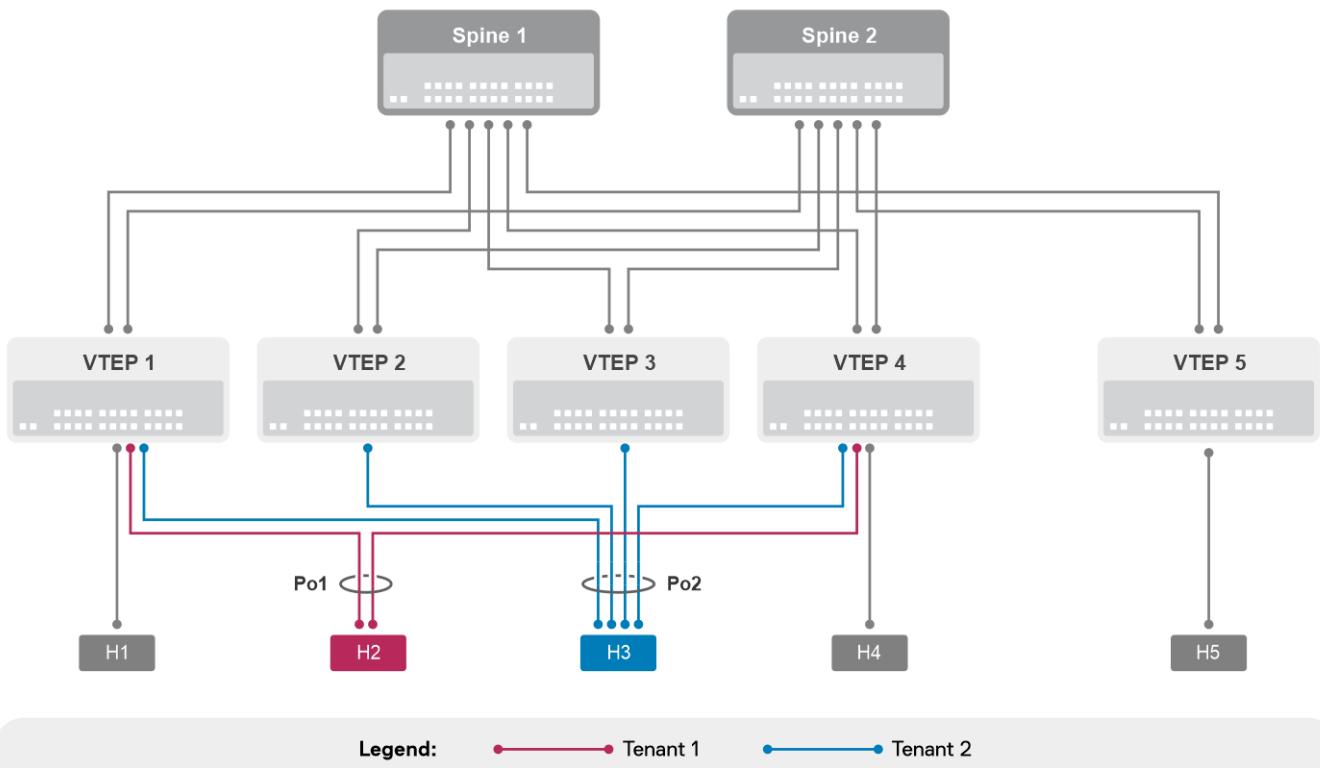
- The system MAC address must be the same on each leaf VTEP in an EVPN Ethernet segment ID.
- The ES-ID is a unique number used to derive the 10-byte EVPN Ethernet segment ID. For a multihomed Ethernet segment, the ES-ID value must be the same on all multihoming VTEPs in a port channel.

In this example, a Type-3 ESI is configured on the port-channel interfaces. The VTEP 1 and VTEP 4 port-channel configurations are:

```
sonic# interface PortChannel 1
sonic(conf-if-po1)# evpn ethernet-segment auto-system-mac
sonic(conf-if-po1)# system-mac 00:00:00:00:11:11
!
sonic# interface PortChannel 2
sonic(conf-if-po2)# evpn ethernet-segment auto-system-mac
sonic(conf-if-po2)# system-mac 00:00:00:00:22:22
```

Whereas on VTEP 2 and VTEP 3, only the PortChannel2 configuration is required because since these VTEPs use only PortChannel2 for EVPN multihoming:

```
sonic# interface PortChannel 2
sonic(conf-if-po2)# evpn ethernet-segment auto-system-mac
sonic(conf-if-po2)# system-mac 00:00:00:00:22:22
```



**Figure 13. EVPN multihoming example**

Sample configurations for VTEPs 1 to 5 are provided here.

VTEPs 1 and 4:

```
sonic(config)# evpn esi-multihoming
sonic(config-evpn-esi-mh)# startup-delay 300
!
sonic(config)# interface PortChannel1
sonic(config-if-po1)# evpn ethernet-segment 00:00:00:00:11:22:33:00:00:01
sonic(config-if-po1)# system-mac 00:00:00:11:22:33
sonic(config-if-po1)# switchport trunk allowed vlan add 100
sonic(config-if-po1)# no shutdown
!
sonic(config)# interface PortChannel2
sonic(config-if-po2)# evpn ethernet-segment auto-system-mac
```

```

sonic(config-if-po2) # system-mac 00:00:00:11:22:33
sonic(config-if-po2) # switchport trunk allowed vlan add 100
sonic(config-if-po2) # no shutdown
!
sonic(config) # interface Eth1/1
sonic(config-if-Eth1/1) # channel-group 1
sonic(config-if-Eth1/1) # no shutdown
!
sonic(config) # interface Eth1/2
sonic(config-if-Eth1/2) # channel-group 2
sonic(config-if-Eth1/2) # no shutdown
!
sonic(config) # interface Loopback1
sonic(config-if-lo1) # ip address 1.1.1.1/32
!
sonic(config) # ip vrf Vrf1
sonic(config) # ip anycast mac-address 00:00:00:0a:0b:0c
sonic(config) # interface Vlan100
sonic(config-if-Vlan100) # ip vrf forwarding Vrf1
sonic(config-if-Vlan100) # ip anycast-address 10.0.0.1/24
!
sonic(config) # interface Vlan1000
sonic(config-if-Vlan1000) # ip vrf forwarding Vrf1
sonic(config-if-Vlan1000) # ipv6 enable
!
sonic(config) # interface vxlan vtep1
sonic(config-if-vtep1) # source-ip 1.1.1.1
sonic(config-if-vtep1) # map vni 100 vlan 100
sonic(config-if-vtep1) # map vni 1000 vlan 1000
sonic(config-if-vtep1) # map vni 1000 vrf Vrf1
!
sonic# config terminal
sonic(config) # link state track uplinkTrack
sonic(config-link-track) # downstream all-evpn-es
sonic(config-link-track) # timeout 180
!
sonic(config) # interface Eth1/5
!! Spine link !!
sonic(config-if-Eth1/5) # ipv6 enable
sonic(config-if-Eth1/5) # link state track uplinkTrack upstream
sonic(config-if-Eth1/5) # no shutdown
!
sonic(config) # interface Eth1/7
!! Spine link !!
sonic(config-if-Eth1/7) # ipv6 enable
sonic(config-if-Eth1/7) # link state track uplinkTrack upstream
sonic(config-if-Eth1/7) # no shutdown
!
sonic# config terminal
sonic(config) # router bgp 10
sonic(config-router-bgp) # router-id 1.1.1.1
sonic(config-router-bgp) # neighbor interface Eth1/5
sonic(config-router-bgp-neighbor) # address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af) # activate
sonic(config-router-bgp-neighbor-af) # exit
!
sonic(config-router-bgp-neighbor) # address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af) # activate
sonic(config-router-bgp-neighbor-af) # exit
sonic(config-router-bgp-neighbor) # exit
!
sonic(config-router-bgp) # neighbor interface Eth1/7
sonic(config-router-bgp-neighbor) # address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af) # activate
sonic(config-router-bgp-neighbor-af) # exit
!
sonic(config-router-bgp-neighbor) # address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af) # activate
sonic(config-router-bgp-neighbor-af) # exit
sonic(config-router-bgp-neighbor) # exit
!
sonic(config-router-bgp) # address-family ipv4 unicast
sonic(config-router-bgp-af) # redistribute connected

```

```

sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise-all-vni
sonic(config-router-bgp-af)# exit
!
sonic(config)# router bgp 10 vrf Vrf1
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family ipv6 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise ipv4 unicast
sonic(config-router-bgp-af)# advertise ipv6 unicast

```

VTEPs 2 and 3:

```

sonic(config)# evpn esi-multihoming
sonic(config-evpn-esi-mh)# startup-delay 300
!
sonic(config)# interface PortChannel2
sonic(config-if-po2)# evpn ethernet-segment auto-system-mac
sonic(config-if-po2)# system-mac 00:00:00:11:22:33
sonic(config-if-po2)# switchport trunk allowed vlan add 100
sonic(config-if-po2)# no shutdown
!
sonic(config)# interface Eth1/1
sonic(config-if-Eth1/1)# channel-group 1
sonic(config-if-Eth1/1)# no shutdown
!
sonic(config)# interface Eth1/2
sonic(config-if-Eth1/2)# channel-group 2
sonic(config-if-Eth1/2)# no shutdown
!
sonic(config)# interface Loopback1
sonic(config-if-lol1)# ip address 2.2.2.2/32
!
sonic(config)# ip vrf Vrf1
sonic(config)# ip anycast mac-address 00:00:00:0a:0b:0c
sonic(config)# interface Vlan100
sonic(config-if-Vlan100)# ip vrf forwarding Vrf1
sonic(config-if-Vlan100)# ip anycast-address 10.0.0.1/24
!
sonic(config)# interface Vlan1000
sonic(config-if-Vlan1000)# ip vrf forwarding Vrf1
sonic(config-if-Vlan1000)# ipv6 enable
!
sonic(config)# interface vxlan vtep2
sonic(config-if-vtep1)# source-ip 2.2.2.2
sonic(config-if-vtep1)# map vni 100 vlan 100
sonic(config-if-vtep1)# map vni 1000 vlan 1000
sonic(config-if-vtep1)# map vni 1000 vrf Vrf1
!
sonic# config terminal
sonic(config)# link state track uplinkTrack
sonic(config-link-track)# downstream all-evpn-es
sonic(config-link-track)# timeout 180
!
sonic(config)# interface Eth1/5
!! Spine link !!
sonic(config-if-Eth1/5)# ipv6 enable
sonic(config-if-Eth1/5)# link state track uplinkTrack upstream
sonic(config-if-Eth1/5)# no shutdown
!
sonic(config)# interface Eth1/7
!! Spine link !!
sonic(config-if-Eth1/7)# ipv6 enable
sonic(config-if-Eth1/7)# link state track uplinkTrack upstream
sonic(config-if-Eth1/7)# no shutdown

```

```

!
sonic# config terminal
sonic(config)# router bgp 10
sonic(config-router-bgp)# router-id 2.2.2.2
sonic(config-router-bgp)# neighbor interface Eth1/5
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
!
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
sonic(config-router-bgp-neighbor)# exit
!
sonic(config-router-bgp)# neighbor interface Eth1/7
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
!
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
sonic(config-router-bgp-neighbor)# exit
!
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise-all-vni
sonic(config-router-bgp-af)# exit
sonic(config-router-bgp)# exit
!
sonic(config)router bgp 10 vrf Vrf1
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family ipv6 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise ipv4 unicast
sonic(config-router-bgp-af)# advertise ipv6 unicast

```

VTEP 5:

```

sonic(config)# interface Loopback1
sonic(config-if-lol1)# ip address 5.5.5.5/32
!
sonic(config)# ip anycast mac-address 00:00:00:0a:0b:0c
sonic(config)# ip vrf Vrf1
sonic(config)# interface Vlan100
sonic(config-if-Vlan100)# ip vrf forwarding Vrf1
sonic(config-if-Vlan100)# ip anycast-address 10.0.0.1/24
!
sonic(config)# interface Vlan1000
sonic(config-if-Vlan1000)# ip vrf forwarding Vrf1
sonic(config-if-Vlan1000)# ipv6 enable
!
sonic(config)# interface vxlan vtep5
sonic(config-if-vtep2)# source-ip 5.5.5.5
sonic(config-if-vtep2)# map vni 100 vlan 100
sonic(config-if-vtep2)# map vni 1000 vlan 1000
sonic(config-if-vtep2)# map vni 1000 vrf Vrf1
sonic(config-if-vtep2)# exit
!
sonic(config)# interface Eth1/5
sonic(config-if-Eth1/5)# ipv6 enable
sonic(config-if-Eth1/5)# no shutdown
!
sonic(config)# interface Eth1/7

```

```

sonic(config-if-Eth1/7)# ipv6 enable
sonic(config-if-Eth1/7)# no shutdown
!
sonic(config)# router bgp 50
sonic(config-router-bgp# router-id 5.5.5.5
sonic(config-router-bgp)# neighbor interface Ethernet16
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
!
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
sonic(config-router-bgp-neighbor)# exit
!
sonic(config-router-bgp)# neighbor interface Ethernet16
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
!
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
sonic(config-router-bgp-neighbor-af)# exit
sonic(config-router-bgp-neighbor)# exit
!
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise-all-vni
sonic(config-router-bgp-af)# exit
sonic(config-router-bgp)# exit
!
sonic(config)# router bgp 50 vrf Vrf1
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family ipv6 unicast
sonic(config-router-bgp-af)# redistribute connected
sonic(config-router-bgp-af)# exit
!
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise ipv4 unicast
sonic(config-router-bgp-af)# advertise ipv6 unicast

```

### **View EVPN multihoming configuration and operation**

In a successful EVPN multihoming configuration, the following settings are displayed:

- Type: Local, Remote
- State: up
- In a multihomed EVPN Ethernet segment, all remote VTEPs are displayed. The DF status of one of the multi-homed VTEPs is df and is used as the preferred Designated Forwarder. The DF status of the other remote VTEPs is non-df.

Use the show evpn, show bgp l2vpn evpn, and show bgp l2vpn es-evi commands to troubleshoot an EVPN multihoming configuration.

```

sonic# show evpn es detail
ESI: 03:00:00:00:11:22:33:00:00:01
Type: Local, Remote
Interface: PortChannel1
State: up
Bridge port: yes
Ready for BGP: yes
VNI Count: 2
MAC Count: 1
DF status: df
DF preference: 32767
Nexthop group: 536870913
VTEPs:

```

```

4.4.4.4 df_alg: preference df_pref: 32767 nh: 268435458
ESI: 03:00:00:00:11:22:33:00:00:02
Type: Local,Remote
Interface: PortChannel12
State: up
Bridge port: yes
Ready for BGP: yes
VNI Count: 2
MAC Count: 1
DF status: df
DF preference: 32767
Nexthop group: 536870914
VTEPs:
 2.2.2.2 df_alg: preference df_pref: 32767 nh: 268435459
 3.3.3.3 df_alg: preference df_pref: 32767 nh: 268435460
 4.4.4.4 df_alg: preference df_pref: 32767 nh: 268435461

```

```

sonic# show evpn es-evi detail
VNI 200 ESI: 03:00:00:00:11:22:33:00:00:01
 Type: L
 Ready for BGP: yes

VNI 100 ESI: 03:00:00:00:11:22:33:00:00:01
 Type: L
 Ready for BGP: yes

VNI 200 ESI: 03:00:00:00:11:22:33:00:00:02
 Type: L
 Ready for BGP: yes

VNI 100 ESI: 03:00:00:00:11:22:33:00:00:02
 Type: L
 Ready for BGP: yes

```

```

sonic# show evpn l2-nh
VTEP NH id #ES
1.1.1.1 268435462 1
2.2.2.2 268435461 1
3.3.3.3 268435463 3

```

```

sonic# show evpn
L2 VNIs: 6
L3 VNIs: 1
Advertise gateway mac-ip: No
Advertise svi mac-ip: No
Advertise svi mac: No
Duplicate address detection: Enable
 Detection max-moves 5, time 180
EVPN MH:
 mac-holdtime: 1080s, neigh-holdtime: 1080s
 uplink-cfg-cnt: 0, uplink-active-cnt: 0
IPv4 Neigh Kernel threshold: 48000
IPv6 Neigh Kernel threshold: 48000
Total IPv4 neighbors: 2
Total IPv6 neighbors: 3

```

```

sonic# show bgp l2vpn evpn es detail
ESI: 03:00:00:00:11:22:33:00:00:01
 Type: LR
 RD: 1.1.1.1:3
 Originator-IP: 1.1.1.1
 Local ES DF preference: 32767
 VNI Count: 2
 Remote VNI Count: 2
 VRF Count: 1
 MACIP EVI Path Count: 2
 MACIP Global Path Count: 5
 Inconsistent VNI VTEP Count: 0
 Inconsistencies: -

```

```

VTEPs:
 4.4.4.4 flags: EA df_alg: preference df_pref: 32767

ESI: 03:00:00:00:11:22:33:00:00:02
Type: LR
RD: 1.1.1.1:3
Originator-IP: 1.1.1.1
Local ES DF preference: 32767
VNI Count: 2
Remote VNI Count: 2
VRF Count: 1
MACIP EVI Path Count: 2
MACIP Global Path Count: 5
Inconsistent VNI VTEP Count: 0
Inconsistencies: -
VTEPs:
 2.2.2.2 flags: EA df_alg: preference df_pref: 32767
 3.3.3.3 flags: EA df_alg: preference df_pref: 32767
 4.4.4.4 flags: EA df_alg: preference df_pref: 32767

```

```

sonic# show bgp 12vpn evpn es-evi detail
VNI: 100 ESI: 03:00:00:00:11:22:33:00:00:01
Type: LR
Inconsistencies: -
VTEPs: 2.2.2.2(EV)

VNI: 200 ESI: 03:00:00:00:11:22:33:00:00:01
Type: LR
Inconsistencies: -
VTEPs: 2.2.2.2(EV)

```

**i | NOTE:** In `show bgp 12vpn evpn es detail` and `show bgp 12vpn evpn es-evi detail` output, the `Inconsistencies` field displays the misconfigurations across leaf nodes in an EVPN Ethernet segment. To resolve a misconfiguration, check the VXLAN configuration on each VTEP.

```

sonic# show bgp 12vpn evpn es-vrf detail
ES-VRF Flags: A Active
ESI VRF Flags IPv4-NHG IPv6-NHG Ref
00:00:00:00:00:00:00:00:00:00:00 VRF Vrf001 70313757 70313758 16
00:00:00:00:00:00:00:00:00:00:00 VRF Vrf002 70313759 70313760 16
00:00:00:00:00:00:00:00:00:00:00 VRF Vrf003 70313761 70313762 16
00:00:00:00:00:00:00:00:00:00:00 VRF Vrf004 70313763 70313764 16

```

```

sonic# show bgp 12vpn evpn next-hops
VRF IP RMAC #Paths Base Path
RED 10.10.10.3 44:38:39:22:01:bb 4 10.1.10.0/24
RED 10.10.10.4 44:38:39:22:01:c1 4 10.1.10.0/24
RED 10.10.10.2 44:38:39:22:01:af 4 10.1.10.0/24
BLUE 10.10.10.3 44:38:39:22:01:bb 2 10.1.30.0/24
BLUE 10.10.10.4 44:38:39:22:01:c1 2 10.1.30.0/24
BLUE 10.10.10.2 44:38:39:22:01:af 2 10.1.30.0/24

```

```

sonic# show vxlan remote nexthop-group
NHG Remote VTEPs Local Members
== ====== =====
22 2.3.4.5
 3.4.5.6

23 1.1.1.2 PortChannel15
 1.1.1.3

```

```

sonic# show vxlan remote mac

Vlan MAC Type Tunnel Group VNI
==== == === ===== === ===

```

```
10 00:00:00:11:22:33 Dynamic 2.3.4.5 Internal 100
 3.4.5.6
```

```
sonic# show evpn es startup-delay
Startup Delay : 600 secs
Time left : 0 secs
```

## Multichassis LAG

A port channel (LAG) allows you to bundle multiple interfaces together into an aggregate group for redundancy and increased bandwidth. All links are on the same switch. A multichassis LAG (MCLAG) allows you to create a logical switch in which multiple interfaces on peer switches are bundled. The MCLAG peer switches are managed separately as independent devices.

The MCLAG provides redundancy and load balancing between the MCLAG peers. A downstream switch or server connects to the MCLAG peers through a multichassis port channel.

An MCLAG consists of:

- MCLAG domain — Two peer switches connected with a keepalive and a peer link.
  - Keepalive link — Layer 3 link that connects MCLAG peer switches. It carries periodic heartbeat messages between MCLAG peer devices to monitor their operational status. It is also used to synchronize states between MCLAG peer devices.
  - Peer link — Connects MCLAG peer switches and acts as data backup path between MCLAG peers. It is used to carry data traffic when a MCLAG member port is down.
- MCLAG peer switch — The other switch in the MCLAG connected with the keepalive and peer links. One MCLAG peer is selected as active; the other peer is standby. The active and standby roles are determined by switch IP address — the peer with the lowest IP address is selected as active.
- MCLAG interface — Member port interface on one of the peer switches that is assigned to the MCLAG.

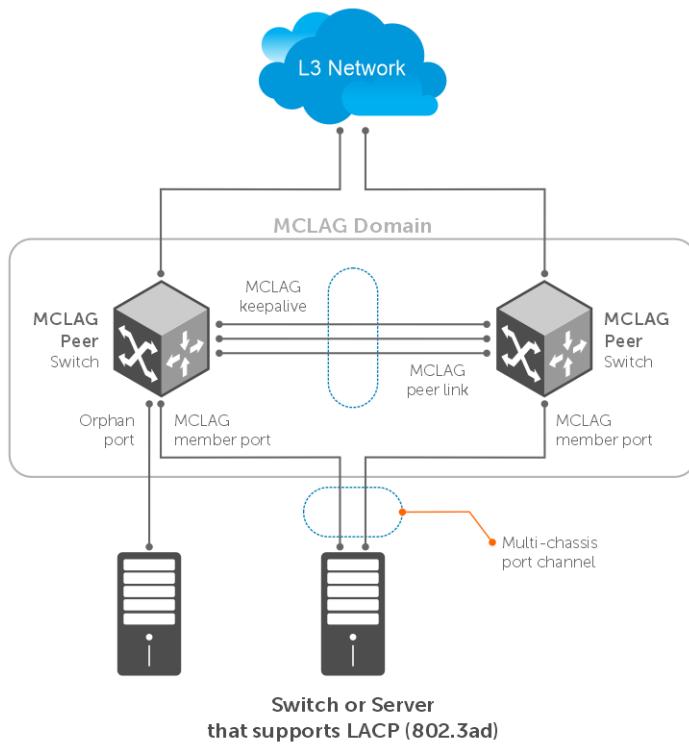
Configure a delay restore timer to hold an MCLAG interface down for a specified time during a switch reboot, system crash, or when you run the `fast-reboot` command. The delay restore feature allows upstream protocols to stabilize traffic while waiting for the MCLAG link to reconverge. As a result, drops in south-to-north traffic are minimized when a peer switch restarts. To activate the delay restore timer, you must enable link tracking.

On an MCLAG switch, port interfaces that are not configured as member interfaces in the MCLAG, and which connect to downstream devices and belong to VLANs synchronized through the peer link, are *orphan ports*.

**i** **NOTE:** A VRRP active-active configuration is not supported in an MCLAG. Use a static anycast gateway instead by configuring an IP anycast address on MCLAG peer switches.

### MCLAG with VRF

MCLAG supports the default VRF, Management VRF, or a nondefault VRF.



**Figure 14. MCLAG topology**

### Topics:

- MCLAG configuration
- View MCLAG configuration
- MCLAG peer gateway
- Troubleshoot MCLAG

## MCLAG configuration

1. On each MCLAG peer, create an MCLAG domain and enter domain configuration mode. Valid domain numbers are 1 to 4095. Only one MCLAG domain is supported on a switch.

```
sonic(config) # mclag domain domain-id
sonic(conf-mclag-domain) #
```

To unconfigure an MCLAG domain, enter the no `mclag domain domain-id` command.

2. Configure local and remote peer settings in an MCLAG domain. On each MCLAG peer, configure the following:
  - Local IPv4 address — IPv4 address of the MCLAG keepalive link on the local switch. If you enter a loopback interface, it must be configured with an IPv4 address.
  - Peer IPv4 address — IPv4 address of the MCLAG keepalive link on the peer switch.
  - VRF for the session between MCLAG peers — The VRF can be the Management VRF, the default VRF, or a non-default VRF. Before you configure the session VRF (`session-vrf` command) with a non-default VRF, create the VRF. If you do not use a session-specific VRF, the default VRF is used. You must bind the MCLAG keepalive link to the session VRF.
  - Local port channel or peer-link interface that connects to the peer. Configure the peer-link interface on all VLANs in an MCLAG peer.
  - Unique MCLAG system MAC address that is assigned to the MCLAG domain on both peers. To prevent MCLAG port channel flapping on the standby peer when the active peer reboots, Dell Technologies recommends that you configure a system MAC address. If you do not configure a system MAC address, the MAC address of the active peer is used as the MCLAG MAC address across both peers.
  - Delay restore timer (1 to 3600 seconds; default 300) — Starts when the first IP interfaces come up. Use the delay restore timer to minimize south-to-north traffic drops during MCLAG startup. The timer avoids traffic loss by holding

down MLAG interfaces and orphan ports (physical ports or MLAG port channel) while waiting for protocols to converge.

```
sonic(config)# mclag domain domain-id
sonic(conf-mclag-domain)# source-ip {local-peer-ipv4-address | Loopback number}
sonic(conf-mclag-domain)# peer-ip remote-peer-ipv4-address
sonic(conf-mclag-domain)# session-vrf vrf-name
sonic(conf-mclag-domain)# peer-link {PortChannelmclag-portchannel-number | Eth slot/
port}
sonic(conf-mclag-domain)# mclag-system-mac local-peer-mac-address
sonic(conf-mclag-domain)# delay-restore seconds
```

- For the `session-vrf vrf-name` option, enter `mgmt` for Management VRF, or the name of the nondefault VRF.
3. Configure MLAG session parameters. On each MLAG peer, configure the following:
- Time interval between sending keepalive messages over the peer link to determine if the remote peer is up or down (1 to 60 seconds, default 1).
  - Time to wait before shutting down an MLAG session with a remote peer if no keepalive reply is received (1 to 3600 seconds, default 30). Enter a session-timeout value that is equal to a multiple number of keepalive intervals.

```
sonic(conf-mclag-domain)# keepalive-interval seconds
sonic(conf-mclag-domain)# session-timeout seconds
```

4. On each MLAG peer, configure the member interfaces in port channels between the MLAG and an attached device; for example:

```
sonic(config)# interface Eth1/28
sonic(conf-if-Eth1/28)# description Server2
sonic(conf-if-Eth1/28)# channel-group 255
sonic(conf-if-Eth1/28)# exit

sonic(config)# interface Eth1/29
sonic(conf-if-Eth1/29)# description Server2
sonic(conf-if-Eth1/29)# channel-group 255
sonic(conf-if-Eth1/29)# exit

sonic(config)# interface port-channel255
sonic(conf-if-po255)# description mclagtoserver
sonic(conf-if-po255)# mclag domain-id
sonic(conf-if-po255)# exit
sonic(config) #
```

5. (Optional) By default, an MLAG switch and its peer use the system MAC address of the active peer as the gateway MAC address in L3 interfaces. To use a common configurable gateway MAC address for the L3 VLAN interfaces in which the peer link is a VLAN member, configure an MLAG gateway MAC address.

```
sonic(config)# mclag gateway-mac xx:xx:xx:xx:xx:xx
```

6. (Optional) Enable link tracking for downstream links — see [Link state tracking](#). Use link state tracking to allow traffic from downstream links to be sent to an MLAG peer if all upstream links on the local switch are down. Downstream MLAG interfaces on the local switch are shut down. Upstream links on the MLAG peer transmit traffic to the spine. Link state tracking avoids the need to increase bandwidth on the peer link to handle the additional upstream traffic.

```
sonic(config)# link state track gourp-name
sonic(conf-link-track)# downstream all-mclag
sonic(conf-link-track)# exit
```

7. (Optional) To enable L3 protocols, such as BGP, OSPF, and BFD, to transmit packets on VLANs between MLAG peers, configure each peer VLAN interface using the `mclag-separate-ip` command. By default, VLANs on both MLAG peers use the same MAC and IP addresses, which prevents L3 protocols to pass traffic between them.

```
sonic(config)# interface Vlan vlan-id
sonic(conf-if-Vlan)# mclag-separate-ip
```

To verify the VLANs on an MLAG peer that are configured for IP routing, use the `show mclag separate-ip-interfaces` command; for example:

```
sonic# show mclag separate-ip-interfaces
Interface Name
=====
```

```
Vlan4094
=====
Total count : 1
=====
```

To verify MLAG configuration, use the `show mclag brief` command; for example:

```
sonic# show mclag brief

Domain ID : 1
Role : active
Session Status : up
Peer Link Status :
Source Address : 100.104.78.38
Peer Address : 100.104.78.39
Session Vrf : mgmt
Peer Link : Eth1/13
Keepalive Interval : 1 secs
Session Timeout : 30 secs
Delay Restore : 300 secs
System Mac : 3c:2c:30:85:db:04
Mclag System Mac : 00:01:01:01:01:01

Number of MLAG Interfaces:1

MLAG Interface Local/Remote Status

PortChannel10 up/up
```

**(i) NOTE:** On an MLAG L3-provisioned interface, multiple secondary IPv4 subnet configurations are not supported; for example:

```
sonic(config)# interface Vlan51
sonic(conf-if-Vlan)# neigh-suppress
sonic(conf-if-Vlan)# ip vrf forwarding Vrf-IPT
sonic(conf-if-Vlan)# ip anycast-address 66.148.65.1/24
sonic(conf-if-Vlan)# ip anycast-address 66.148.64.1/24
sonic(conf-if-Vlan)# ip anycast-address 66.235.178.1/24
```

#### Example: MLAG configuration with default VRF on peer switches

**Table 36. MLAG configuration with default VRF on peer switches**

| Peer 1                                                                                                                                                                                    | Peer 2                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ! Create a spanned VLAN across peers.                                                                                                                                                     |                                                                                                                                                                                           |
| sonic(config)# interface Vlan101<br>sonic(conf-if-Vlan101)# exit                                                                                                                          | sonic(config)# interface Vlan101<br>sonic(conf-if-Vlan101)# exit                                                                                                                          |
| ! Create an MLAG LACP port channel and associate it with each VLAN that spans the port channel.                                                                                           |                                                                                                                                                                                           |
| sonic(config)# interface PortChannel 1<br>fast_rate<br>sonic(conf-if-pol1)# switchport trunk<br>allowed Vlan add 101-105<br>sonic(conf-if-pol1)# no shutdown<br>sonic(conf-if-pol1)# exit | sonic(config)# interface PortChannel 1<br>fast_rate<br>sonic(conf-if-pol1)# switchport trunk<br>allowed Vlan add 101-105<br>sonic(conf-if-pol1)# no shutdown<br>sonic(conf-if-pol1)# exit |
| ! Configure the orphan port in VLAN101.                                                                                                                                                   | N/A                                                                                                                                                                                       |
| sonic(config)# interface Eth1/12<br>sonic(conf-if-Eth1/12)# switchport trunk<br>allowed Vlan add 101<br>sonic(conf-if-Eth1/12)# no shutdown<br>sonic(conf-if-Eth1/12)# exit               |                                                                                                                                                                                           |

**Table 36. MLAG configuration with default VRF on peer switches (continued)**

| Peer 1                                                                                                                                                                                                                                                                                                                                                                                                 | Peer 2                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ! Configure the member interface in MLAG port channel1.                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>sonic(config)# interface Eth1/11 sonic(conf-if-Eth1/11)# channel-group 1 sonic(conf-if-Eth1/11)# no shutdown sonic(conf-if-Eth1/11)# exit</pre>                                                                                                                                                                                                                                                   | <pre>sonic(config)# interface Eth1/11 sonic(conf-if-Eth1/11)# channel-group 1 sonic(conf-if-Eth1/11)# no shutdown sonic(conf-if-Eth1/11)# exit</pre>                                                                                                                                                                                                                                                   |
| ! Configure the peer-link port channel.                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>sonic(config)# interface PortChannel 100 sonic(conf-if-po100)# switchport trunk allowed Vlan add 101 sonic(conf-if-po100)# no shutdown sonic(conf-if-po100)# exit</pre>                                                                                                                                                                                                                           | <pre>sonic(config)# interface PortChannel 100 sonic(conf-if-po100)# switchport trunk allowed Vlan add 101 sonic(conf-if-po100)# no shutdown sonic(conf-if-po100)# exit</pre>                                                                                                                                                                                                                           |
| ! Assign interface to peer-link port channel.                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>sonic(config)# interface Eth1/73 sonic(conf-if-Eth1/73)# channel-group 100 sonic(conf-if-Eth1/73)# no shutdown</pre>                                                                                                                                                                                                                                                                              | <pre>sonic(config)# interface Eth1/77 sonic(conf-if-Eth1/77)# channel-group 100 sonic(conf-if-Eth1/77)# no shutdown</pre>                                                                                                                                                                                                                                                                              |
| ! Configure the MLAG domain.                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>sonic(config)# mclag domain 1 sonic(conf-mclag-domain-1)# source-ip 192.168.100.2 sonic(conf-mclag-domain-1)# peer-ip 192.168.100.3 sonic(conf-mclag-domain-1)# peer-link PortChannel100 sonic(conf-mclag-domain-1)# mclag-system- mac 00:00:00:11:11:11 sonic(conf-mclag-domain-1)# keepalive- interval 1 sonic(conf-mclag-domain-1)# session- timeout 30 sonic(conf-mclag-domain-1)# exit</pre> | <pre>sonic(config)# mclag domain 1 sonic(conf-mclag-domain-1)# source-ip 192.168.100.3 sonic(conf-mclag-domain-1)# peer-ip 192.168.100.2 sonic(conf-mclag-domain-1)# peer-link PortChannel100 sonic(conf-mclag-domain-1)# mclag-system- mac 00:00:00:11:11:11 sonic(conf-mclag-domain-1)# keepalive- interval 1 sonic(conf-mclag-domain-1)# session- timeout 30 sonic(conf-mclag-domain-1)# exit</pre> |
| ! Configure the MLAG port channel.                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>sonic(config)# interface PortChannel 1 sonic(conf-if-po1)# mclag 1 sonic(conf-if-po1)# exit sonic(config) #</pre>                                                                                                                                                                                                                                                                                 | <pre>sonic(config)# interface PortChannel 1 sonic(conf-if-po1)# mclag 1 sonic(conf-if-po1)# exit sonic(config) #</pre>                                                                                                                                                                                                                                                                                 |

**Example: MLAG configuration on peer switches with non-default or Management VRF**

** NOTE:**

- The configured session VRF is used only to establish the MLAG session.
- An MLAG domain does not bind dual-homed MLAG interfaces to the configured session VRF. To bind a dual-homed MLAG interface to a non-default VRF or management VRF, see [Virtual routing and forwarding](#).

**Table 37. MLAG configuration on peer switches with non-default or Management VRF**

| Peer 1                                | Peer 2 |
|---------------------------------------|--------|
| ! Create a spanned VLAN across peers. |        |

**Table 37. MCLAG configuration on peer switches with non-default or Management VRF (continued)**

| Peer 1                                                                                                                                                                             | Peer 2                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>sonic(config)# interface Vlan101 sonic(conf-if-Vlan101)# exit</pre>                                                                                                           | <pre>sonic(config)# interface Vlan101 sonic(conf-if-Vlan101)# exit</pre>                                                                                                           |
| ! Create an MCLAG LACP port channel and associate it with each VLAN that spans the port channel.                                                                                   |                                                                                                                                                                                    |
| <pre>sonic(config)# interface PortChannel 1 fast_rate sonic(conf-if-po1)# switchport trunk allowed Vlan add 101-105 sonic(conf-if-po1)# no shutdown sonic(conf-if-po1)# exit</pre> | <pre>sonic(config)# interface PortChannel 1 fast_rate sonic(conf-if-po1)# switchport trunk allowed Vlan add 101-105 sonic(conf-if-po1)# no shutdown sonic(conf-if-po1)# exit</pre> |
| ! Configure the orphan port in VLAN101.                                                                                                                                            | N/A                                                                                                                                                                                |
| <pre>sonic(config)# interface Eth1/12 sonic(conf-if-Eth1/12)# switchport trunk allowed Vlan add 101 sonic(conf-if-Eth1/12)# no shutdown sonic(conf-if-Eth1/12)# exit</pre>         |                                                                                                                                                                                    |
| ! Configure the member interface in MCLAG port channel1.                                                                                                                           |                                                                                                                                                                                    |
| <pre>sonic(config)# interface Eth1/11 sonic(conf-if-Eth1/11)# channel-group 1 sonic(conf-if-Eth1/11)# no shutdown sonic(conf-if-Eth1/11)# exit</pre>                               | <pre>sonic(config)# interface Eth1/11 sonic(conf-if-Eth1/11)# channel-group 1 sonic(conf-if-Eth1/11)# no shutdown sonic(conf-if-Eth1/11)# exit</pre>                               |
| ! Configure the peer-link port channel.                                                                                                                                            |                                                                                                                                                                                    |
| <pre>sonic(config)# interface PortChannel 100 sonic(conf-if-po100)# switchport trunk allowed Vlan add 101 sonic(conf-if-po100)# no shutdown sonic(conf-if-po100)# exit</pre>       | <pre>sonic(config)# interface PortChannel 100 sonic(conf-if-po100)# switchport trunk allowed Vlan add 101 sonic(conf-if-po100)# no shutdown sonic(conf-if-po100)# exit</pre>       |
| ! Assign interface to a peer-link port channel.                                                                                                                                    |                                                                                                                                                                                    |
| <pre>sonic(config)# interface Eth1/73 sonic(conf-if-Eth1/73)# channel-group 100 sonic(conf-if-Eth1/73)# no shutdown</pre>                                                          | <pre>sonic(config)# interface Eth1/77 sonic(conf-if-Eth1/77)# channel-group 100 sonic(conf-if-Eth1/77)# no shutdown</pre>                                                          |
| ! Create a VRF.                                                                                                                                                                    |                                                                                                                                                                                    |
| <pre>sonic(config)# ip vrf VRF-Red sonic(config)# sonic(config) # mclag domain 1 sonic(config-mclag-domain-1)# session-vrf VRF-Red</pre>                                           | <pre>sonic(config)# ip vrf VRF-Red sonic(config)# sonic(config) # mclag domain 1 sonic(config-mclag-domain-1)# session-vrf VRF-Red</pre>                                           |
| Or                                                                                                                                                                                 | Or                                                                                                                                                                                 |
| <pre>sonic(config)# ip vrf mgmt sonic(config)# sonic(config) # mclag domain 1 sonic(config-mclag-domain-1)# session-vrf mgmt</pre>                                                 | <pre>sonic(config)# ip vrf mgmt sonic(config)# sonic(config) # mclag domain 1 sonic(config-mclag-domain-1)# session-vrf mgmt</pre>                                                 |
| ! Bind the peer keepalive link to the configured session VRF. <b>Note:</b> This configuration is optional when the Management VRF is used.                                         |                                                                                                                                                                                    |

**Table 37. MLAG configuration on peer switches with non-default or Management VRF (continued)**

| Peer 1                                                                                                                                                                                                                                                                                                                                                                                                 | Peer 2                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>sonic(config)# interface Eth1/1 sonic(config-if-Eth1/1)# ip vrf forwarding VrfRed sonic(config-if-Eth1/1)# ip address 192.168.100.2/24 sonic(config-if-Eth1/1)# exit</pre>                                                                                                                                                                                                                        | <pre>sonic(config)# interface Eth1/1 sonic(config-if-Eth1/1)# ip vrf forwarding VrfRed sonic(config-if-Eth1/1)# ip address 192.168.100.3/24 sonic(config-if-Eth1/1)# exit</pre>                                                                                                                                                                                                                        |
| ! Configure the MLAG domain.                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>sonic(config)# mclag domain 1 sonic(conf-mclag-domain-1)# source-ip 192.168.100.2 sonic(conf-mclag-domain-1)# peer-ip 192.168.100.3 sonic(conf-mclag-domain-1)# peer-link PortChannel100 sonic(conf-mclag-domain-1)# mclag-system- mac 00:00:00:11:11:11 sonic(conf-mclag-domain-1)# keepalive- interval 1 sonic(conf-mclag-domain-1)# session- timeout 30 sonic(conf-mclag-domain-1)# exit</pre> | <pre>sonic(config)# mclag domain 1 sonic(conf-mclag-domain-1)# source-ip 192.168.100.3 sonic(conf-mclag-domain-1)# peer-ip 192.168.100.2 sonic(conf-mclag-domain-1)# peer-link PortChannel100 sonic(conf-mclag-domain-1)# mclag-system- mac 00:00:00:11:11:11 sonic(conf-mclag-domain-1)# keepalive- interval 1 sonic(conf-mclag-domain-1)# session- timeout 30 sonic(conf-mclag-domain-1)# exit</pre> |
| ! Configure the MLAG port channel.                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                        |
| <pre>sonic(config)# interface PortChannel 1 sonic(conf-if-po1)# mclag 1 sonic(conf-if-po1)# exit sonic(config) #</pre>                                                                                                                                                                                                                                                                                 | <pre>sonic(config)# interface PortChannel 1 sonic(conf-if-po1)# mclag 1 sonic(conf-if-po1)# exit sonic(config) #</pre>                                                                                                                                                                                                                                                                                 |

## View MLAG configuration

### Peer 1

```
sonic# show mclag brief
Domain ID : 1
Role : active
Session Status : up
Peer Link Status : up
Source Address : 192.168.100.2
Peer Address : 192.168.100.3
Session Vrf : default
Peer Link : PortChannel100
Keepalive Interval : 1 secs
Session Timeout : 30 secs
Delay Restore : 300 secs
System Mac : 8c:04:ba:cb:c8:40
Mclag System Mac : 00:00:00:11:11:11
Number of MLAG Interfaces: 1

MLAG Interface Local/Remote Status

PortChannel1 up/up

sonic# show PortChannel summary
Flags(oper-status): D - Down U - Up (portchannel) P - Up in portchannel (members)

Group PortChannel Type Protocol Member Ports

1 PortChannel1 (U) Eth LACP Eth1/11(P)
```

|     |                    |     |      |                            |
|-----|--------------------|-----|------|----------------------------|
| 100 | PortChannel100 (U) | Eth | LACP | Eth1/72 (D)<br>Eth1/77 (P) |
|-----|--------------------|-----|------|----------------------------|

## Peer 2

```

sonic# show mclag brief
Domain ID : 1
Role : standby
Session Status : up
Peer Link Status : up
Source Address : 192.168.100.3
Peer Address : 192.168.100.2
Session Vrf : default
Peer Link : PortChannel100
Keepalive Interval : 1 secs
Session Timeout : 30 secs
Delay Restore : 300 secs
System Mac : 8c:04:ba:cb:c8:40
Mclag System Mac : 00:00:00:11:11:11
Number of MLAG Interfaces: 1

MLAG Interface Local/Remote Status

PortChannel1 up/up

sonic# show PortChannel summary
Flags(oper-status): D - Down U - Up (portchannel) P - Up in portchannel (members)

Group PortChannel Type Protocol Member Ports

1 PortChannel11 (U) Eth LACP Eth1/11 (P)
100 PortChannel100 (U) Eth LACP Eth1/72 (D)
 Eth1/77 (P)

```

## View status of MCLAG peer link

To verify the status of the MCLAG peer link, use the `show mclag interface mclag-portchannel-number domain-id` command.

```

sonic# show mclag interface 1 1
Local/Remote Status : up/up
TrafficDisable : No
IsolateWithPeerLink : Yes

```

# MCLAG peer gateway

An MCLAG peer gateway allows an MCLAG node to act as an active gateway for packets that are addressed to the router MAC address of the MCLAG peer node. This functionality is useful when MCLAG clients do not send ARP requests to identify the default gateway.

In a Layer 3 MCLAG topology, when an MCLAG router routes a packet to the MCLAG client, the reply packet that the MCLAG client sends may reach either the MCLAG node that originally sent the packet or the peer MCLAG node due to LAG hashing. If the MCLAG peer node receives the reply packet, it switches the frames to the MCLAG node that originally routed the packet. The packet takes a longer route to reach the destination, resulting in a suboptimal routing.

When you enable the MCLAG peer gateway functionality on both MCLAG peers, the MCLAG node that receives a reply packet acts as the gateway and routes the packet to the destination without switching the frame to its MCLAG peer.

### Limitations for MCLAG peer gateway

- OSPF between an MCLAG node and an MCLAG client is not supported.
- BFD single hop sessions between an MCLAG node and an MCLAG client are not supported.

### Configuration notes

- Configure a unique IP address on each VLAN interface on the MCLAG peers.
- Enable the MCLAG peer gateway functionality on both MCLAG peers.

### Configure MCLAG peer gateway

1. Configure MCLAG. For more details on configuring MCLAG, see [MCLAG configuration](#).
2. Configure a separate IP address on VLAN interface for L3 protocol support over MCLAG:

```
mclag-separate-ip
```

3. Enable the MCLAG peer gateway functionality on the VLAN interfaces:

```
mclag-peer-gateway
```

4. Configure a unique IP address on each VLAN interface.

```
ip address unique-ip-address
```

#### **View MCLAG peer gateway information**

```
sonic# show mclag peer-gateway
Interface Name
=====
Vlan10
=====
Total count : 1
=====
```

## Troubleshoot MCLAG

To troubleshoot MCLAG operation, log in as a root user to the Linux shell and enter this command:

```
root@sonic:/home/admin# mclagdctl dump debug counters
ICCP session down: 1
Peer link down: 0
Rx invalid msg: 0
Rx sock error(hdr): 0
Rx zero len(hdr): 0
Rx zero len(tlv): 0
Rx sock error(tlv): 0
Rx zero len(tlv): 0
Rx hdr retry max: 0
Rx hdr retry total: 0
Rx hdr retry fail: 0
Rx retry max: 0
Rx retry total: 0
Rx retry fail: 0
Rx fail zero len: 0
Rx fail error: 0
Rx stp fail zero len:0
Rx stp fail error 0
Socket close err: 0
Socket cleanup: 0
Warmboot: 0

ICCP to MclagSyncd TX_OK TX_ERROR
----- ----- -----
PortIsolation 6 0
MacLearnMode 0 0
FlushFdb 1 0
SetIfMac 0 0
SetFdb 0 0
SetL2mc 0 0
TrafficDistEnable 1 0
TrafficDistDisable 1 0
SetIccpState 2 0
SetIccpRole 1 0
SetSystemId 0 0
DelIccpInfo 0 0
SetRemoteIntfSts 10 0
DelRemoteIntf 0 0
SetL2mcMrouter 0 0
PeerLinkIsolation 10 0
LacpFallback 20 0
```

|                                 |       |          |          |          |
|---------------------------------|-------|----------|----------|----------|
| SetPeerSystemId                 | 1     | 0        |          |          |
| SetMclagSystemId                | 1     | 0        |          |          |
| SetMclagLocalIfMac              | 2     | 0        |          |          |
| SetMclagPoDisable               | 0     | 0        |          |          |
| SetMclagPoEnable                | 0     | 0        |          |          |
| <br>MclagSyncd to ICCP          | RX_OK | RX_ERROR |          |          |
| -----                           | ----- | -----    |          |          |
| FdbChange                       | 0     | 0        |          |          |
| CfgMclag                        | 2     | 0        |          |          |
| CfgMclagIface                   | 2     | 0        |          |          |
| CfgMclagUniqueIp                | 0     | 0        |          |          |
| vlanMbrshipChange               | 9     | 0        |          |          |
| LacpFallback                    | 51    | 0        |          |          |
| CfgMclagGWMac                   | 0     | 0        |          |          |
| <br>ICCP to Peer                | TX_OK | RX_OK    | TX_ERROR | RX_ERROR |
| -----                           | ----- | -----    | -----    | -----    |
| SysConfig                       | 1     | 1        | 0        | 0        |
| AggrConfig                      | 1     | 1        | 0        | 0        |
| AggrState                       | 3     | 4        | 0        | 0        |
| MacInfo                         | 0     | 0        | 0        | 0        |
| ArpInfo                         | 295   | 287      | 0        | 0        |
| L2mcInfo                        | 0     | 0        | 0        | 0        |
| PoInfo                          | 1     | 1        | 0        | 0        |
| ArpInfo                         | 295   | 287      | 0        | 0        |
| L2mcInfo                        | 0     | 0        | 0        | 0        |
| PoInfo                          | 1     | 1        | 0        | 0        |
| PeerLinkInfo                    | 1     | 1        | 0        | 0        |
| Heartbeat                       | 44866 | 44865    | 0        | 0        |
| Nak                             | 0     | 0        | 0        | 0        |
| SyncData                        | 2     | 2        | 0        | 0        |
| SyncReq                         | 1     | 0        | 0        | 0        |
| Warmboot                        | 0     | 0        | 0        | 0        |
| IfUpAck                         | 2     | 1        | 0        | 0        |
| StpConnect                      | 0     | 0        | 0        | 0        |
| StpDisconnect                   | 0     | 0        | 0        | 0        |
| StpNrpvUpd                      | 0     | 0        | 0        | 0        |
| StpNrpvReq                      | 0     | 0        | 0        | 0        |
| StpRootPortReq                  | 0     | 0        | 0        | 0        |
| StpRootPortAck                  | 0     | 0        | 0        | 0        |
| StpTc                           | 0     | 0        | 0        | 0        |
| StpAgeOut                       | 0     | 0        | 0        | 0        |
| StpMasterReq                    | 0     | 0        | 0        | 0        |
| StpMasterResp                   | 0     | 0        | 0        | 0        |
| StpPortInfo                     | 0     | 0        | 0        | 0        |
| StpPropSyncReq                  | 0     | 0        | 0        | 0        |
| StpPropSyncResp                 | 0     | 0        | 0        | 0        |
| LacpFlbk                        | 5     | 5        | 0        | 0        |
| NDiscInfo                       | 0     | 0        | 0        | 0        |
| StpBridgeInfo                   | 0     | 0        | 0        | 0        |
| StpMclagInfo                    | 0     | 0        | 0        | 0        |
| <br>Netlink Counters            |       |          |          |          |
| -----                           |       |          |          |          |
| Link add/del:                   | 384/0 |          |          |          |
| Unknown if_name:                | 7     |          |          |          |
| Not AF_BRIDGE:                  | 0     |          |          |          |
| Neighbor(ARP) add/del:          | 256/0 |          |          |          |
| MAC entry add/del:              | 0/0   |          |          |          |
| Address add/del:                | 1/0   |          |          |          |
| Unexpected message type:        | 0     |          |          |          |
| Receive error:                  | 0     |          |          |          |
| Send error:                     | 0     |          |          |          |
| PO MAC addr change err:         | 0     |          |          |          |
| <br>Memory alloc/free Counters  |       |          |          |          |
| MAC entry allocated count:      | 0     |          |          |          |
| MAC entry free count:           | 0     |          |          |          |
| ARP entry RB allocated count:   | 1     |          |          |          |
| ARP entry RB free count:        | 0     |          |          |          |
| ARP entry List allocated count: | 295   |          |          |          |
| ARP entry List free count:      | 295   |          |          |          |

```
ND entry RB allocated count: 0
ND entry RB free count: 0
ND entry List allocated count: 0
ND entry List free count: 0
```

## Access control lists

Access control lists (ACLs) filter inbound and outbound traffic passing through the switch using match criteria. The match criteria correspond to Layer 2, 3 and 4 parameters in a packet header. For example, you can specify source IP addresses to permit or deny packets, and add upper-level TCP or UDP protocol rules for additional filtering. Use ACLs for security to limit switch access and for packet classification in flow-based software features, such as QoS and PBR.

### Topics:

- [ACLs for security and packet classification](#)
- [ACLs for flow-based services](#)
- [Policy-based replication groups](#)
- [ACL consistency checker](#)
- [Enable ACL counters](#)
- [View ACL configuration and counters](#)

## ACLs for security and packet classification

Use ACLs to restrict access to a switch interface by using rules that permit and deny L2 and L3 traffic. ACLs provide security and packet classification for flow-based features, including:

- Block unwanted traffic or users from accessing the network resources.
- Save network resources by reducing unwanted traffic.
- Reduce the chance of denial-of-service (DOS) attacks.
- Classify traffic for QoS actions, such as rate limiting, PCP remarking in VLAN headers, DSCP remarking, and so on.
- Classify traffic for packet forwarding, such as policy-based routing (PBR).

Layer 2 ACLs filter traffic based on MAC header fields, such as Source MAC, Destination MAC, and VLAN ID. Use L2 ACLs to filter traffic of any EtherType, including IPv4 and IPv6. Apply L2 ACLs globally or on any Ethernet, port channel, or VLAN interface.

Global ACLs filter all traffic that is bridged or routed on switch interfaces. Global switch-level ACLs support MAC, IPv4, and IPv6 ACLs.

ACLs applied on an Ethernet or port channel interface process L2 and L3 packets to determine whether to forward or drop a packet based on the permit/deny criteria in the ACL.

ACLs applied to a VLAN filter all traffic that is bridged within the same VLAN, or that is routed in or out of a VLAN. Apply VLAN ACLs to both bridged and routed traffic. VLAN ACLs support MAC, IPv4, and IPv6 ACLs.

ACLs are processed in sequential order from the first to the last numbered entry. When a match is found, no further ACL processing is performed.

By default, all L2 MAC, IPv4, and IPv6 ACLs contain a `deny any` rule at the end. The `deny any` rule drops all traffic that does not match preceding permit/deny entries in the ACL. The default `deny any` rule is not applicable in ACLs used for flow-based services (see [ACLs for flow-based services](#)).

Add a `permit any any` rule to the end of an ACL to permit all packets that are not denied by other criteria.

**i | NOTE:** Add a `permit any any` rule only if you want to permit any unmatched traffic.

To configure an ACL and apply it on an interface:

1. Create an ACL to filter Layer 2, IPv4, or IPv6 traffic.
2. Add MAC, IPv4, or IPv6 permit/deny rules to the ACL.
3. Apply the ACL on ingress or egress interfaces, or on the control plane.

# Configure ACLs

Create an ACL to filter L2, IPv4, or IPv6 traffic on ingress and egress interfaces. Add permit and deny rules for IP/IPv6 traffic, and for packets containing TCP, UDP, and ICMP protocol values.

1. Create a L2 MAC, L3 IPv4, or IPv6 ACL. The ACL name must begin with an alphanumeric character (up to 63 characters).

```
sonic(config)# {mac | ip | ipv6} access-list name
```

To delete the ACL, enter the no version of the command.

2. Add a permit/deny rules to allow or drop frames/packets from a source and to a destination device. Reenter the command to enter additional rules. Rules are applied in sequentially numbered order. To delete an ACL rule, enter no seq 1-65535.

- **MAC ACL rule**

```
sonic(conf-mac-acl)# seq 1-65535 {permit | deny} {any | host source-mac-address [/source-mac-mask]} {any | host dest-mac-address | dest-mac-mask} [ethertype | ipv4 | ipv6 | arp] [vlan vlan-id] [pcp 0-7] [dei 0-1] remark description
```

- seq <1-65535> — Numbered order in which the permit/deny statement is processed in the ACL.
- any — Permit or deny all source/destination addresses.
- host source-mac-address — Source MAC address in xxxx.xxxx.xxxx or xx:xx:xx:xx:xx:xx format
- /source-mac-mask — Source MAC address mask in xxxx.xxxx.xxxx or xx:xx:xx:xx:xx:xx format
- host dest-mac-address — Destination MAC address in xxxx.xxxx.xxxx or xx:xx:xx:xx:xx:xx format
- /dest-mac-mask — Destination MAC address mask in xxxx.xxxx.xxxx or xx:xx:xx:xx:xx:xx format
- ethertype — Permit or deny Ethernet frames with an EtherType value (1536 to 65536). Enter the EtherType value in hexadecimal or decimal format.
- ipv4 | ipv6 | arp — Permit or deny IPv4, IPv6, or ARP packets. A MAC ACL also applies to L3 traffic that is processed.
- vlan vlan-id — Permit or deny a VLAN ID (1 to 4094).
- pcp — Permit or deny an Ethernet frame with a Priority Code Point (PCP) value (0 to 7) in the header.
- dei — Permit or deny an Ethernet frame with a Drop Eligible Indicator (DEI) value (0 or 1) in the header.
- remark description — Text that describes the rule; 256 characters maximum. If the text contains blank spaces, enclose the text in double quotes (").

- **IPv4 ACL rules — IP, TCP, UDP, ICMP**

```
sonic(conf-ipv4-acl)# seq 1-65535 {permit | deny} {ip | ip-protocol} {any | host source-ip-address [/source-ip-prefix-len]} {any | host dest-ip-address [/dest-ip-prefix-len]} [dscp dscp-value] [vlan vlan-id] [remark description]
```

```
sonic(conf-ipv4-acl)# seq 1-65535 {permit | deny} tcp {any | host source-ip-address [/source-ip-prefix-len]} [{eq | gt | lt} tcp-port-number | range begin end] {any | host dest-ip-address [/dest-ip-prefix-len]} [{eq | gt | lt} tcp-port-number | range begin end] [dscp dscp-value] [established | [fin | not-fin] [syn | not-syn] [rst | not-rst] [psh | not-psh] [ack | not-ack] [urg | not-urg]] [vlan vlan-id] [remark description]
```

```
sonic(conf-ipv4-acl)# seq 1-65535 {permit | deny} udp {any | host source-ip-address [/source-ip-prefix-len]} [{eq | gt | lt} udp-port-number | range begin end] {any | host dest-ip-address [/dest-ip-prefix-len]} [{eq | gt | lt} udp-port-number | range begin end] [dscp dscp-value] [vlan vlan-id] [remark description]
```

```
sonic(conf-ipv4-acl)# seq 1-65535 {permit | deny} icmp {any | host source-ip-address [/source-ip-prefix-len]} {any | host dest-ip-address [/dest-ip-prefix-len]} [dscp dscp-value] [type icmp-type] [code icmp-code] [vlan vlan-id] [remark description]
```

- seq 1-65535 — Numbered order in which the permit/deny statement is processed in the ACL.
- ip | ip-protocol — Permit or deny IP packets or packets from an IP protocol:
  - Internet control message protocol (ICMP) — 1
  - Internet group membership protocol (IGMP) — 2
  - Transmission control protocol (TCP) — 6

- User datagram protocol (UDP) — 17
- Resource reservation protocol (RSVP) — 46
- Generic routing encapsulation (GRE) — 47
- Authentication header (AUTH) — 51
- Protocol-independent multicast (PIM) — 103
- Layer two tunneling protocol v.3 (L2TP) — 115
- any — Permit or deny all IP source/destination addresses.
- host *source-ip-address* — Source IP address in dotted decimal format *A.B.C.D[/mask]*, where *mask* is an IPv4 prefix-mask number (0 to 32): 10.10.10.0/24.
- host *dest-ip-address* — Destination IP address in dotted decimal *A.B.C.D[/mask]* format.
- eq | gt | lt — TCP or UDP port numbers that are equal to greater than, or less than the port number that follows.
- *port-number* | range *begin end* — TCP or UDP port number, or range of port numbers; valid only when the *ip-protocol* is 6 (TCP) or 17 (UDP).
- dscp *value* — Permit or deny packets with DSCP value (0 to 63).
- vlan *vlan-id* — Permit or deny a VLAN ID (1 to 4094).
- remark *description* — Text that describes the rule. If the text contains blank spaces, enclose the text in double quotes (").
- established | [fin | not-fin] [syn | not-syn] [rst | not-rst] [psh | not-psh] [ack | not-ack] [urg | not-urg] — TCP flags applied when the *ip-protocol* is 6 (TCP). established and other TCP flags are mutually exclusive. established is equivalent to matching the ack or rst flags. A TCP not-flag matches the flag bit 0. If you specify multiple TCP flags, all flags must match a TCP packet header.
- type *icmp-type* — ICMP type (0 to 255) that is applied when the *ip-protocol* is 1 (ICMP).
- code *icmp-code* — ICMP code (0 to 255) that is applied when the *ip-protocol* is 1 (ICMP).

- **IPv6 ACL rules — IPv6, TCP, UDP, ICMP**

```
sonic(conf-ipv4-acl)# seq 1-65535 {permit | deny} ipv6-protocol {any | host source-ipv6-address [/source-ipv6-prefix-len]} {any | host dest-ipv6-address [/dest-ipv6-prefix-len]} [dscp dscp-value] [vlan vlan-id] [remark description]
```

```
sonic(conf-ipv4-acl)# seq 1-65535 {permit | deny} tcp {any | host source-ipv6-address [/source-ipv6-prefix-len]} [{eq | gt | lt} tcp-port-number | range begin end] {any | host dest-ipv6-address [/dest-ipv6-prefix-len]} [{eq | gt | lt} tcp-port-number | range begin end] [dscp dscp-value] [established | [fin | not-fin] [syn | not-syn] [rst | not-rst] [psh | not-psh] [ack | not-ack] [urg | not-urg]] [vlan vlan-id] [remark description]
```

```
sonic(conf-ipv4-acl)# seq 1-65535 {permit | deny} udp {any | host source-ipv6-address [/source-ipv6-prefix-len]} [{eq | gt | lt} udp-port-number | range begin end] {any | host dest-ipv6-address [/dest-ipv6-prefix-len]} [{eq | gt | lt} udp-port-number | range begin end] [dscp dscp-value] [vlan vlan-id] [remark description]
```

```
sonic(conf-ipv4-acl)# seq 1-65535 {permit | deny} icmp {any | host source-ipv6-address [/source-ipv6-prefix-len]} {any | host dest-ipv6-address [/dest-ipv6-prefix-len]} [dscp dscp-value] [type icmp-type] [code icmp-code] [vlan vlan-id] [remark description]
```

- seq <1-65535> — Numbered order in which the permit/deny statement is processed in the ACL.
- *ipv6-protocol* — Permit or deny packets from an IPv6 protocol:
  - TCP — 6
  - UDP — 17
  - ICMPv6 — 58
  - IPv6 — any
- any — Permit or deny all IPv6 source/destination addresses.
- host *source-ipv6-address* — Source IPv6 address in the format *x:x:x:x:x:x:x:/mask*, where *mask* is an IPv6 prefix-mask number (0 to 128); for example, 2001:db8:1234:0000::/64.
- host *dest-ipv6-address* — Destination IPv6 address in the format *x:x:x:x:x:x:x:/mask*.

- `eq | gt | lt` — TCP or UDP port numbers that are equal to greater than, or less than the port number that follows.
  - `port-number | range begin end` — TCP or UDP port number (0 to 65535), or range of port numbers; valid only when the `ip-protocol` is 6 (TCP) or 17 (UDP).
  - `dscp value` — Permit or deny packets with DSCP value (0 to 63).
  - `vlan vlan-id` — Permit or deny a VLAN ID (1 to 4094).
  - `remark description` — Text that describes the rule. If the text contains blank spaces, enclose the text in double quotes ("").
  - `established | [fin | not-fin] [syn | not-syn] [rst | not-rst] [psh | not-psh] [ack | not-ack] [urg | not-urg]` — TCP flags applied when the `ip-protocol` is 6 (TCP). `established` and other TCP flags are mutually exclusive. `established` is equivalent to matching `ack` or `rst` flags. A TCP `not-` flag matches the flag bit value 0. If you specify multiple TCP flags, all flags must match in a TCP packet header.
  - `type icmp-type` — ICMP type (0 to 255) that is applied when the `ip-protocol` is 58 (ICMPv6).
  - `code icmp-code` — ICMP code (0 to 255) that is applied when the `ip-protocol` is 58 (ICMPv6).
3. Apply a MAC, IPv4, or IPv6 ACL on an ingress or egress interface, or on all ingress or egress interfaces on the switch. To remove the ACL, enter the `no` version of the command.

```
sonic(config)# interface Eth slot/port
sonic(conf-if-xxx)# {mac | ip | ipv6} access-group acl-name {in | out}
```

Or

```
sonic(config)# {mac | ip | ipv6} access-group acl-name {in | out}
```

4. (Optional) Apply an IPv4 or IPv6 ACL on the control plane. To remove the ACL, enter the `no` version of the command.

**i|NOTE:** A VTY ACL applies to data plane traffic in addition to the management control plane.

```
sonic(config)# line vty
sonic(conf-line-vty)# {mac | ip | ipv6} access-group name {in | out}
```

## L2 and L3 ACL interaction

When a packet matches a statement in a L2 MAC or a L3 IPv4 or IPv6 security ACL, one of these actions is taken:

- Permit — Packet is forwarded in the data plane; a trap is sent to the CPU.
- Deny — Packet is dropped in the data plane; a trap is sent to the CPU.
- Transit — Packet is forwarded in the data plane; a trap is not sent to the CPU.
- Discard — Packet is dropped in the data plane; a trap is not sent to the CPU.

If you apply L2 MAC and L3 IPv4 or IPv6 ACLs on the same interface, incoming packets can match statements in both L2 and L3 ACLs. ACL counters are increased to record the match in both ACLs. If either the L2 or IP/IPv6 ACL is not configured, not matched, or not applicable, only the result from the matched/applied ACL is applied. The following table shows the possible results when both a L2 and L3 ACL are applied to incoming traffic.

**i|NOTE:** If both L2 and L3 ACLs are applied, the L2 ACL result also applies to the L3 traffic that is processed.

**Table 38. L2 and L3 ACL interaction**

| L2 ACL result | L3 ACL result | Packet forwarding allowed | Trap to CPU allowed |
|---------------|---------------|---------------------------|---------------------|
| Permit        | Permit        | Yes                       | Yes                 |
| Permit        | Deny          | No                        | Yes                 |
| Permit        | Transit       | Yes                       | No                  |
| Permit        | Discard       | No                        | No                  |
| Deny          | Permit        | No                        | Yes                 |
| Deny          | Deny          | No                        | Yes                 |
| Deny          | Transit       | No                        | No                  |

**Table 38. L2 and L3 ACL interaction (continued)**

| <b>L2 ACL result</b> | <b>L3 ACL result</b> | <b>Packet forwarding allowed</b> | <b>Trap to CPU allowed</b> |
|----------------------|----------------------|----------------------------------|----------------------------|
| Deny                 | Discard              | No                               | No                         |
| Transit              | Permit               | Yes                              | No                         |
| Transit              | Deny                 | No                               | No                         |
| Transit              | Transit              | Yes                              | No                         |
| Transit              | Discard              | No                               | No                         |
| Discard              | Permit               | No                               | No                         |
| Discard              | Deny                 | No                               | No                         |
| Discard              | Transit              | No                               | No                         |
| Discard              | Discard              | No                               | No                         |

## Security ACL examples

### MAC ACL configuration

```
sonic(config)# mac access-list macacl
sonic(conf-mac-acl)# seq 1 permit 0000.1000.0000 0000.ffff.0000 any
sonic(conf-mac-acl)# seq 2 permit any any ip
sonic(conf-mac-acl)# seq 3 permit any any ipv6
sonic(conf-mac-acl)# seq 4 deny any 0100.0000.0000 0100.0000.0000
```

### IPv4 ACL configuration

```
sonic(config)# ip access-list ipacl
sonic(conf-ipv4-acl)# seq 1 permit tcp 10.0.0.0/8 any
sonic(conf-ipv4-acl)# seq 2 deny udp any 20.1.0.0/16 gt 1024
sonic(conf-ipv4-acl)# seq 3 deny ip any any dscp 63
```

### IPv6 ACL configuration

```
sonic(config)# ipv6 access-list ipv6acl
sonic(conf-ipv6-acl)# seq 1 permit ipv6 1000::/16 any dscp 20
sonic(conf-ipv6-acl)# seq 2 deny tcp any 2000::1000:0/112 range 100 1000
sonic(conf-ipv6-acl)# seq 3 permit tcp any any established
sonic(conf-ipv6-acl)# seq 4 deny udp any eq 3000 any
```

## ACLs for flow-based services

ACLs provide a common infrastructure for use in flow-based services, such as:

- QoS packet remarking and policing — see [Quality of Service](#).
- SPAN and ERSPAN monitoring — see [Port monitoring](#).
- PBR and L2 redirect to route/forward packets to a specified next hop or to bypass a recommended path - see [Policy-based routing](#).
- CoPP to rate limit CPU-bound traffic and set the CPU trap queue — see [Control plane policing](#).

ACLs allow you to filter and reset L2 and L3 parameters to better control, and fine-tune traffic flows by using match and set statements. Match rules classify incoming packets using fields in L2, L3, and L4 headers. Set actions are performed on matching packets.

ACLs are identified by name, and process match statements in sequence. If a packet does not match the criterion in the first match entry, the second match entry applies, then the third and so on.

Flow-based services on a switch use modular ACL configuration to identify and process traffic in a flow. To configure and apply ACLs on flow-based traffic:

1. Create a class map to identify the traffic to be processed.

2. Create a policy map to specify the configuring actions to take on classified traffic.
3. Apply the policy to an ingress or egress (if supported) interface.

## Configure flow-based ACLs

Use ACLs to classify traffic flows based on their permit/deny statements. Configure the classified traffic in a policy for a flow-based service, such as QoS, monitoring, forwarding, or on the CPU interface. In the policy, configure the actions to be taken on matching traffic.

1. Create a class map to identify a traffic flow. A class-map can use either an ACL to match the incoming traffic or fields from L2, L3, and L4 headers. The traffic is selected using a MAC, IP, or IPv6 ACL — see [Configure ACLs](#). The class-map name must begin with an alphanumeric character (up to 63 characters). It can contain alphanumeric, hyphen (-), and underscore (\_) characters.

```
sonic(config)# class-map name match-type acl
```

To delete the class map, enter the `no class-map name` command.

- Add a L2 MAC, IPv4, or IPv6 ACL to select flow traffic.

```
sonic(conf-class-map)# match access-group {mac | ip | ipv6} acl-name
```

- Add a class-map description (up to 256 characters).

```
sonic(conf-class-map)# description text
```

2. To create a class map that uses match statements for L2, L3, and L4 header fields to select traffic in a flow, use the `class-map name match-type fields match-all` command.

```
sonic(config)# class-map name match-type fields match-all
```

- Add a class-map description (up to 256 characters).

```
sonic(conf-class-map)# description text
```

- Add any of the following match statements to classify traffic in a `match-type fields` class map.

- Match a source MAC address.

```
sonic(conf-class-map)# match source-address mac source-mac-address [/source-mac-mask]
```

- Match a destination MAC address.

```
sonic(conf-class-map)# match destination-address mac dest-mac-address [/dest-mac-mask]
```

- Match an EtherType value in the Ethernet frame. Enter a user-defined `ethertype` value (1536 to 65536) in hexadecimal or decimal format.

```
sonic(conf-class-map)# match ether-type {ip | ipv6 | ethertype}
```

- Match a PCP value in the Ethernet frame header.

```
sonic(conf-class-map)# match pcp {be | bk | ee | ca | vi | vo | ic | nc | pcp-value}
```

- `be` — Best effort (0)
- `bk` — Background (1)
- `ee` — Excellent effort (2)
- `ca` — Critical applications (3)
- `vi` — Video, less than 100-millisecond latency and jitter (4)
- `vo` — Voice, less than 10-millisecond latency and jitter (5)
- `ic` — Internetwork control (6)
- `nc` — Network control (7)
- `pcp-value` — PCP numeric value (0 to 7)

- Match a VLAN ID.

```
sonic(conf-class-map) # match vlan vlan-id
```

- Match a source IPv4 address.

```
sonic(conf-class-map) # match source-address ip {host ip-address | ip-address/prefix}
```

- Match a destination IPv4 address.

```
sonic(conf-class-map) # match destination-address ip {host ip-address | ip-address/prefix}
```

- Match a source IPv6 address.

```
sonic(conf-class-map) # match source-address ipv6 {host ipv6-address | ipv6-address/prefix}
```

- Match a destination IPv6 address.

```
sonic(conf-class-map) # match source-address ipv6 {host ipv6-address | ipv6-address/prefix}
```

- Match an IP protocol.

```
sonic(conf-class-map) # match ip protocol {tcp | udp | icmp | icmpv6 | ip-protocol-number}
```

- Match a source TCP or UDP port when the *ip-protocol-number* is 6 (TCP).

```
sonic(conf-class-map) # match 14-port source {eq port-number | range begin end}
```

- Match a destination TCP or UDP port when the *ip-protocol-number* is 6 (TCP).

```
sonic(conf-class-map) # match 14-port destination {eq port-number | range begin end}
```

- Match TCP flags when the *ip-protocol-number* is 6 (TCP). Enter a *not-xxx* value to match a TCP flag that is set to 0.

```
sonic(conf-class-map) # match tcp-flags {syn | not-syn} {ack | not-ack} {fin | not-fin} {ack | not-ack} {psh | not-psh} {urg | not-urg}
```

3. Create a policy map to configure the actions to take on classified traffic for a flow-based feature. The policy-map name must begin with an alphanumeric character; 63 characters maximum. It can contain alphanumeric, hyphen (-), and underscore (\_) characters.

```
sonic(config) # policy-map name type (qos | monitoring | forwarding | acl-copp | copp)
```

To delete the policy map, enter the `no policy-map name` command.

- Add a policy-map description (up to 256 characters).

```
sonic(conf-policy-map) # description text
```

- Add a class-map flow to the policy and enter policy-map flow configuration mode. Enter a priority number (0 to 4095) to specify the order in which a class map is applied in the policy map to match traffic in the flow.

```
sonic(conf-policy-map) # class class-map-name priority number
sonic(conf-policy-map-flow) #
```

4. In policy-map-flow mode for a flow-based feature, configure the actions to take on classified traffic. For more information, see these chapters:

- Quality of Service — see [Configure and apply QoS policies](#).
- Monitoring — see [Flow-based port monitoring](#).
- Policy-based routing — see [Configure and apply PBR forwarding policies](#).
- Control plane policing — see [Control plane policing](#).

5. Apply a QoS, monitoring, or forwarding policy map globally on all switch interfaces, on a specified interface, on the control plane, or on the CPU interface. To remove a policy from an interface, enter the no version of the command.

** NOTE:**

- You can apply QoS policies on ingress or egress interfaces.
  - You can apply monitoring and forwarding policies only on ingress interfaces.
  - On the control plane, you can only apply a QoS policy.
  - On the CPU interface, you can only apply an ACL-CoPP policy.
- Globally on all switch interfaces:

```
sonic(config)# service-policy type qos {in | out} policy-map-name
sonic(config)# service-policy type {monitoring | forwarding} in policy-map-name
```

- On an interface or subinterface:

```
sonic(conf-if-xxx)# service-policy type qos {in | out} policy-map-name
sonic(conf-if-xxx))# service-policy type {monitoring | forwarding} in policy-map-name
```

```
sonic(conf-subintf-xxx)# service-policy type qos {in | out} policy-map-name
sonic(conf-subintf-xxx))# service-policy type {monitoring | forwarding} in policy-map-name
```

- On the control plane:

```
sonic(conf-line-vty)# service-policy type qos in policy-map-name
```

- On the CPU:

```
sonic(conf-if-CPU)# service-policy type acl-copp in policy-map-name
```

### Applying policies across different interface types

You can apply a flow-based service policy at the port, VLAN, and switch level. If a packet matches the classified traffic in more than one policy, policy map actions are applied to classified traffic in this order of priority:

1. CPU
2. Port interface or port channel
3. VLAN
4. Switch

If a packet matches a policy that is applied to a port, a VLAN, and the switch, only the port policy is applied.

Because a policy applied to the CPU port has the highest priority, configure specific match criteria for CPU-bound traffic to prevent masking port, port channel, and switch-level policies.

## Flow-based ACL examples

This example shows how to use MAC and IP ACLs in class-maps to select L2 and L3 traffic in QoS, monitoring, and forwarding flows. Each policy map sets the QoS, monitoring, or forwarding actions to take on the selected traffic in the flow.

```
Create L2 MAC ACL
sonic(config)# mac access-list l2_ACL_0
sonic(conf-mac-acl)# seq 1 permit any any ip
sonic(conf-mac-acl)# seq 2 permit any any ipv6

Create IP ACL
sonic(config)# ip access-list l3_ACL_0
sonic(conf-ipv4-acl)# seq 1 permit tcp any any
sonic(conf-ipv4-acl)# seq 2 permit udp any any

Create classifier class0 for IPv4 traffic
sonic(config)# class-map class0 match-type acl
sonic(conf-class-map)# match access-group ip l3_ACL_0

Create classifier class1 for L2 MAC traffic
sonic(config)# class-map class1 match-type acl
```

```

sonic(conf-class-map)# match access-group mac 12_ACL_0

Create policy policy0 for QoS actions

sonic(config)# policy policy0 type qos

Create flow using classifier class0 and set results
sonic(conf-policy-map)# class class0 priority 200
sonic(conf-policy-map-flow)# set pcp 5
sonic(conf-policy-map-flow)# set dscp 15

Create flow using classifier class0 and set results
sonic(conf-policy-map)# class class1 priority 100
sonic(conf-policy-map-flow)# police cir 10mbps cbs 20MB pir 50mbps pbs 100MB

Create policy policy1 for Monitoring actions

sonic(config)# policy policy1 type monitoring

Create flow using class1 and set results
sonic(conf-policy-map)# class class1 priority 100
sonic(conf-policy-map-flow)# set mirror-session test_session

Create policy policy2 for Forwarding actions

sonic(config)# policy policy2 type forwarding
sonic(conf-policy-map)# class class0 priority 100
sonic(conf-policy-map-flow)# set ip next-hop 10.1.1.1 priority 900
sonic(conf-policy-map-flow)# set ip next-hop 100.1.1.1 vrf default priority 800
sonic(conf-policy-map-flow)# set ip next-hop 132.45.2.100 vrf VrfOrange priority 700
sonic(conf-policy-map-flow)# set ip next-hop 100.10.20.30
sonic(conf-policy-map-flow)# set interface null

Apply Qos, monitoring, and forwarding policies on interfaces

sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1)# service-policy type qos in policy0
sonic(conf-if-Eth1/1)# service-policy type monitoring in policy1
sonic(conf-if-Eth1/1)# service-policy type forwarding in policy2
sonic(conf-if-Eth1/1)# exit

sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# service-policy type monitoring in policy1
sonic(conf-if-Eth1/2)# exit

sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# service-policy type forwarding in policy2

```

## Policy-based replication groups

**(i) NOTE:** Policy-based replication using ACLs is available only in the Enterprise Standard and Enterprise Premium bundles. It is not available in the Cloud Standard, Cloud Premium, and Edge Standard bundles.

Policy-based replication provides a mechanism to replicate (copy) a packet to multiple ECMP and different next-hop paths. This feature is designed to resolve the Frame Too Big (FTB) hashing problem described in RFC7690. Use replication groups to replicate traffic to anycast servers.

**(i) NOTE:** The number of packet copies generated by a replication group depends on the unique paths through which all the next-hops are reachable, and not the number of configured next-hops. For example, if a next-hop is reachable through four ECMP paths, four copies of the packet are generated. By default, one packet copy is generated for each reachable path if

single-copy is not enabled. If more than one next-hop is reachable using the same path, only one copy is generated for the path.

A replication group can consist of members and paths which have different speeds.

### Configure and apply a policy-based replication group

1. Create a policy-based forwarding (PBF) replication group. The PBF replication group is used in a policy map that you assign to interfaces. The type entry specifies if the next-hop addresses of all member entries are IPv4 or IPv6. The type entry is optional in the future if you access the replication group to configure additional entries. To delete a PBF replication group, enter the no pbf replication-group name command.

```
sonic(config)# pbf replication-group name type {ip | ipv6}
```

2. Add a replication group entry by specifying a next-hop address. Each entry is used by egress interfaces to forward traffic. Enter a unique ID and next-hop for each entry. Adding duplicate next-hops for different entry IDs is not supported. By default, the next-hop address is reachable by any means: recursive, non-recursive, or overlay. Re-enter the command to add multiple next-hop addresses for replicated packets.

```
sonic(config-pbf-next-hop-group)# entry entry-id next-hop ip-address [vrf vrf-name] [recursive | non-recursive | overlay] [single-copy]
```

- *entry-id* — An ID number from 1 to 65535.
  - *ip-address* — IPv4 or IPv6 address of the next-hop reachable over an underlay or VXLAN overlay network; maximum 128 next-hop addresses. IPv6 member next-hops are not supported in an IPv4 replication group, and vice versa.
  - *vrf vrf-name* — (Optional) VRF used to reach the next-hop address. By default, the VRF to which an interface belongs is used. By default, the VRF of the interface to which the policy is applied, is used.
  - *recursive* — (Optional) The next-hop IP address is in a route in the routing table and may be in a subnet that is not directly connected to the switch.
  - *non-recursive* — (Optional) The next-hop IP address is directly connected to the switch.
  - *overlay* — (Optional) The next-hop IP address is reachable through a VXLAN tunnel.
  - *single-copy* — (Optional) Disables packet replication on multiple paths so that only one copy is sent to an ECMP-reachable path.
3. Create a forwarding policy map and enter policy-map configuration mode. The policy-map name must begin with an alphanumeric character; 63 characters maximum. It can contain alphanumeric, hyphen (-), and underscore (\_) characters.

```
sonic(config)# policy-map name type forwarding
```

To delete the policy map, enter the no policy-map name command.

- Add a policy-map description (up to 256 characters).

```
sonic(conf-policy-map)# description text
```

- Add a class map or ACL to select a traffic flow — see [Configure flow-based ACLs](#).

```
sonic(config)# class-map name match-type acl
```

Or

```
sonic(conf-class-map)# match access-group {mac | ip | ipv6} acl-name
```

- Add a replication group of next-hop addresses to the policy. Enter an optional priority number (0 to 4095) to specify the order in which the replication group is applied in the policy map.

```
sonic(conf-policy-map)# set {ip | ipv6} replication-group name [priority number]
```

To remove the replication group, enter the no set {ip | ipv6} replication-group name command.

4. Apply the replication-group policy globally on all switch interfaces or on a specified interface. To remove a policy from an interface, enter the no version of the command.

- Globally on all switch interfaces:

```
sonic(config)# service-policy type forwarding in policy-map-name
```

- On an interface or subinterface:

```
sonic(conf-if-xxx) # service-policy type forwarding in policy-map-name
```

```
sonic(conf-subintf-xxx) # service-policy type forwarding in policy-map-name
```

#### **Example: Use policy-based replication groups**

```
sonic(config)# pbf replication-group rg-ipv4-anycast-1 type ip
sonic(config)# entry 1 next-hop 100.100.1.1
sonic(config)# pbf replication-group rg-ipv6-anycast-1 type ip
sonic(config)# entry 1 next-hop 1000:1000::1:1

sonic(config)# policy-map pmap-ftb-1 type forwarding
sonic(config-policy-map)# class cmap-ipv4-ftb priority 100
sonic(config-policy-map-flow)# set ip replication-group rg-ipv4-anycast-1
sonic(config-policy-map)# class cmap-ipv6-ftb priority 90
sonic(config-policy-map-flow)# set ipv6 replication-group rg-ipv6-anycast-1

sonic(config)# interface Eth1/1
sonic(conf-if-Eth1/1) # service-policy type forwarding in pmap-ftb-1
```

## ACL consistency checker

To verify the consistency of an IPv4, IPv6, or MAC ACL configuration, use the ACL consistency checker. The consistency checker verifies the validity of each ACL entry and reports any errors.

#### **Check ACL consistency**

- Start the ACL verification process on all ACL configurations, all ACL configurations of a specified type — MAC, IPv4, or IPv6, or a specified ACL.

```
sonic# consistency-check start access-list [[mac | ip | ipv6] name]
```

If necessary, stop the ACL verification process.

```
sonic# consistency-check stop access-list
```

#### **View consistency checker output**

```
sonic# show consistency-check status access-list [brief | detail] [errors]
```

- brief — (Optional) Displays the results of a consistency check on ACL entries in all databases.
- detail — (Optional) Displays detailed information about ACL entries in all databases
- errors — (Optional) Displays the consistency errors in ACL entries.

If you start the ACL verification with the `consistency-check start access-list` command, and then enter the `show consistency-check status access-list` command without optional parameters, the consistency of all ACL entries across all databases is checked. Only the final result is returned: Consistent or Inconsistent.

```
sonic# show consistency-checker status access-list
ACL consistency checker status: Consistent
```

```
sonic# show consistency-checker status access-list
ACL consistency checker status: Inconsistent
```

Use the optional parameters to view more consistency checker results. For example, for a brief overview of ACL consistency across databases:

```
sonic# show consistency-checker status access-list brief
ACL consistency status for macacl-test
=====
Seq Binding AppDB ASICDB HW
```

```
=====
10 Eth1/1 IC IC IC
10 Eth1/2 C C IC
20 Eth1/1 C C IC
20 Eth1/2 C C C
...
(C) Data is consistent
(IC) Data is Inconsistent
```

To view general information about only inconsistent (IC) ACL data:

```
sonic# show consistency-checker status access-list brief errors

ACL consistency status for macacl-test
=====
Seq Binding AppDB ASICDB HW
=====
10 Eth1/1 IC IC IC
10 Eth1/2 C C IC
20 Eth1/1 C C IC
...
(C) Data is consistent
(IC) Data is Inconsistent
```

To display detailed information about the errors found in inconsistent ACL entries, specify `detail errors` in the command:

```
sonic# show consistency-checker status access-list detail errors

ACL consistency status for ipacl-test
=====
Seq Binding Data AppDB ASICDB HW
=====
10 Eth1/1 SrcIP=10.1.1.1/32 10.1.1.2/32*
20 Eth1/1 SrcIP=10.1.1.1/32 SrcIP+
30 Eth1/1 SrcIP=10.1.1.1/32-
40 Eth1/1 Not Found

Extra entries in AppDB: 1
Entry 1 for port Eth1/1:
DstIP=200.1.1.0/24
Seq=15
Action=Deny

Extra entries in ASIC DB: 1
Entry 1 for port Eth1/1:
DstIP=200.1.1.0/24
Protocol=TCP
Seq=50
Action=Permit
...
(*) Inconsistent Value
(+) Missing. Should be added
(-) Extra. Should be deleted
```

## Enable ACL counters

You can enable the collection of ACL statistics on a per-interface or per-ACL entry level.

```
sonic(config)# hardware
sonic(conf-hardware)# access-list
sonic(conf-hardware-acl)# counters {per-entry | per-interface-entry}
```

- `per-entry` — Collects ACL statistics on all interfaces for each ACL permit/deny entry (default).
- `per-interface-entry` — Collects ACL statistics on each interface for each permit/deny entry in the ACLs applied on the interface.

To display ACL counters, use the `show {mac | ip | ipv6} access-list` command.

# View ACL configuration and counters

## View ACL-interface binding

```
sonic# show {mac | ip | ipv6} access-group
```

```
sonic# show ip access-group
Ingress IP access-list ipacl on Eth1/1
Ingress IP access-list ipacl on PortChannel1
Ingress IP access-list ipacl on Vlan100
```

## View ACL configuration and counters

**i** **NOTE:** You can view an ACL configuration on an interface only if you enabled the collection of ACL statistics on a per-interface basis using the counters per-interface-entry command in Hardware-ACL mode (see [Enable ACL counters](#)).

```
sonic# show {mac | ip | ipv6} access-list [acl-name] [interface {Eth slot/port[/breakout-port][.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch}]
```

```
sonic# show ip access-list
ip access-list ipacl
 seq 1 permit ip host 10.1.1.1 host 20.1.1.1 (0 packets) [0 bytes]
 seq 2 permit ip host 10.1.1.2 host 20.1.1.2 (0 packets) [0 bytes]
 seq 3 permit ip host 10.1.1.3 host 20.1.1.3 (0 packets) [0 bytes]
 seq 4 permit ip host 10.1.1.4 host 20.1.1.4 (0 packets) [0 bytes]
```

## Clear ACL counters

```
sonic# clear {mac | ip | ipv6} counters access-list [acl-name]
```

## View class-map configuration

```
sonic# show class-map [class-map-name | match-type {acl | fields}]
```

```
sonic# show class-map class0
Classifier class0 match-type acl
 match-acl 13_ACL_0
 Referenced in flows:
 policy policy0 at priority 200
```

```
sonic# show class-map match-type fields
Classifier fields_class_0 match-type fields
 Description:
 Match:
 src-ip 40.1.1.100/32
 Referenced in flows:
 policy mon_policy_0 at priority 999
 policy qos_policy_0 at priority 999
```

## View policy-map configuration

```
sonic# show policy-map [policy-map-name | type {qos | monitoring | forwarding | acl-copp}]
```

```
sonic# show policy-map qos_policy_0
Policy qos_policy_0 Type qos
 Description:
 Flow fields_class_0 at priority 999
 Description:
 set-pcp 1
```

```

police cir 10000000 cbs 1000000 pir 0 pbs 0
Flow fields_class_1 at priority 998
Description:
set-pcp 2
police cir 20000000 cbs 2000000 pir 0 pbs 0
Flow fields_class_2 at priority 997
Description:
set-pcp 3
police cir 30000000 cbs 3000000 pir 0 pbs 0
Flow fields_class_3 at priority 996
Description:
set-pcp 4
police cir 40000000 cbs 4000000 pir 0 pbs 0
Applied to:
Eth1/1 at ingress

```

```

sonic# show policy-map type monitoring

Policy mon_policy_0 Type monitoring
Description:
Flow fields_class_0 at priority 999
Description:
mirror-session ERSPAN_DestIP_50.1.1.2
Flow fields_class_1 at priority 998
Description:
mirror-session ERSPAN_DestIP_60.1.1.2
Flow fields_class_2 at priority 997
Description:
mirror-session ERSPAN_DestIP_50.1.1.2
Flow fields_class_3 at priority 996
Description:
mirror-session ERSPAN_DestIP_60.1.1.2
Applied to:
Eth1/1 at ingress

```

### **View policy-interface binding**

```

sonic# show service-policy summary [interface {Eth slot/port[/breakout-port]
[.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch | CPU |]
[type {qos | monitoring | forwarding | acl-copp}]

```

```

sonic# show service-policy summary

Eth1/1
 qos policy qos_policy0 at ingress
 monitoring policy mon_policy_0 at ingress
PortChannel100
 qos policy policy0 at egress
Vlan100
 forwarding policy pbr0 at ingress
CPU
 acl-copp policy copp at ingress

```

### **View policy-interface binding and counters**

**i** **NOTE:** ACL CoPP policies apply only to the CPU interface.

```

sonic# show service-policy {interface {Eth slot/port[/breakout-port][.subinterface] |
PortChannel number[.subinterface] | Vlan vlan-id | Switch [type {qos | monitoring |
forwarding}] | policy-map name [interface {Eth slot/port[/breakout-port][.subinterface] |
PortChannel number[.subinterface] | Vlan vlan-id | Switch}]}

```

Or

```

sonic# show service-policy {interface CPU [type acl-copp] | policy-map name [interface
CPU]}

```

```

sonic# show service-policy interface Eth1/1

```

```

Eth1/1
Policy qos_policy_0 Type qos at ingress
Description:
Flow fields_class_3 at priority 996 (Active)
Description:
set-pcp 4
set-dscp 4
police: cir 40000000 cbs 40000000 pir 0 pbs 0
 type bytes mode color-blind
 operational cir 40000000 cbs 40000000 pir 0 pbs 0
 conformed 0 packets 0 bytes action forward
 exceed 0 frames 0 bytes action forward
 violated 0 frames 0 bytes action drop
Packet matches: 0 frames 0 bytes
Flow fields_class_2 at priority 997 (Active)
Description:
set-pcp 3
set-dscp 3
police: cir 30000000 cbs 30000000 pir 0 pbs 0
 type bytes mode color-blind
 operational cir 30000000 cbs 30000000 pir 0 pbs 0
 conformed 0 packets 0 bytes action forward
 exceed 0 frames 0 bytes action forward
 violated 0 frames 0 bytes action drop
Packet matches: 0 frames 0 bytes

```

```

sonic# show service-policy policy mon_policy_0

Eth1/1
Policy mon_policy_0 Type monitoring at ingress
Description:
Flow fields_class_3 at priority 996 (Active)
Description:
mirror-session ERSPAN_DestIP_60.1.1.2
Packet matches: 0 frames 0 bytes
Flow fields_class_2 at priority 997 (Active)
Description:
mirror-session ERSPAN_DestIP_50.1.1.2
Packet matches: 0 frames 0 bytes
Flow fields_class_1 at priority 998 (Active)
Description:
mirror-session ERSPAN_DestIP_60.1.1.2
Packet matches: 0 frames 0 bytes
Flow fields_class_0 at priority 999 (Active)
Description:
mirror-session ERSPAN_DestIP_50.1.1.2
Packet matches: 0 frames 0 bytes

```

```

sonic# show service-policy interface Vlan 100 type forwarding

Vlan100
Policy pbr_policy_example Type forwarding at ingress
Description:
Flow acl_class_1000 at priority 1000 (Active)
Description:
set ip next-hop 10.1.1.1 vrf default
set ip next-hop 20.1.1.1 vrf VrfRed
set ip next-hop 30.1.1.1 (Selected)
set interface null
Packet matches: 128 frames 128000 bytes
Flow acl_class_999 at priority 999 (Active)
Description:
set ip next-hop 11.1.1.1 vrf default (Selected)
set ip next-hop 21.1.1.1 vrf VrfRed
set ip next-hop 31.1.1.1
set interface null
Packet matches: 0 frames 0 bytes
Flow fields_class_0 at priority 999 (Active)
Description:
set ip next-hop 1111::1 vrf default
set ip next-hop 2222::1 vrf VrfRed (Selected)
set ip next-hop 3333::1

```

```
set interface null
Packet matches: 0 frames 0 bytes
```

#### Clear policy-interface counters

 **NOTE:** ACL CoPP policies apply only to the CPU interface.

```
sonic# clear counters service-policy {interface {Eth slot/port[/breakout-port]
[.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch [type {qos
| monitoring | forwarding}] | policy-map name [interface {Eth slot/port[/breakout-port]
[.subinterface] | PortChannel number[.subinterface] | Vlan vlan-id | Switch}]}}
```

Or

```
sonic# clear counters service-policy {interface CPU [type acl-copp] | policy-map name
[interface CPU]}
```

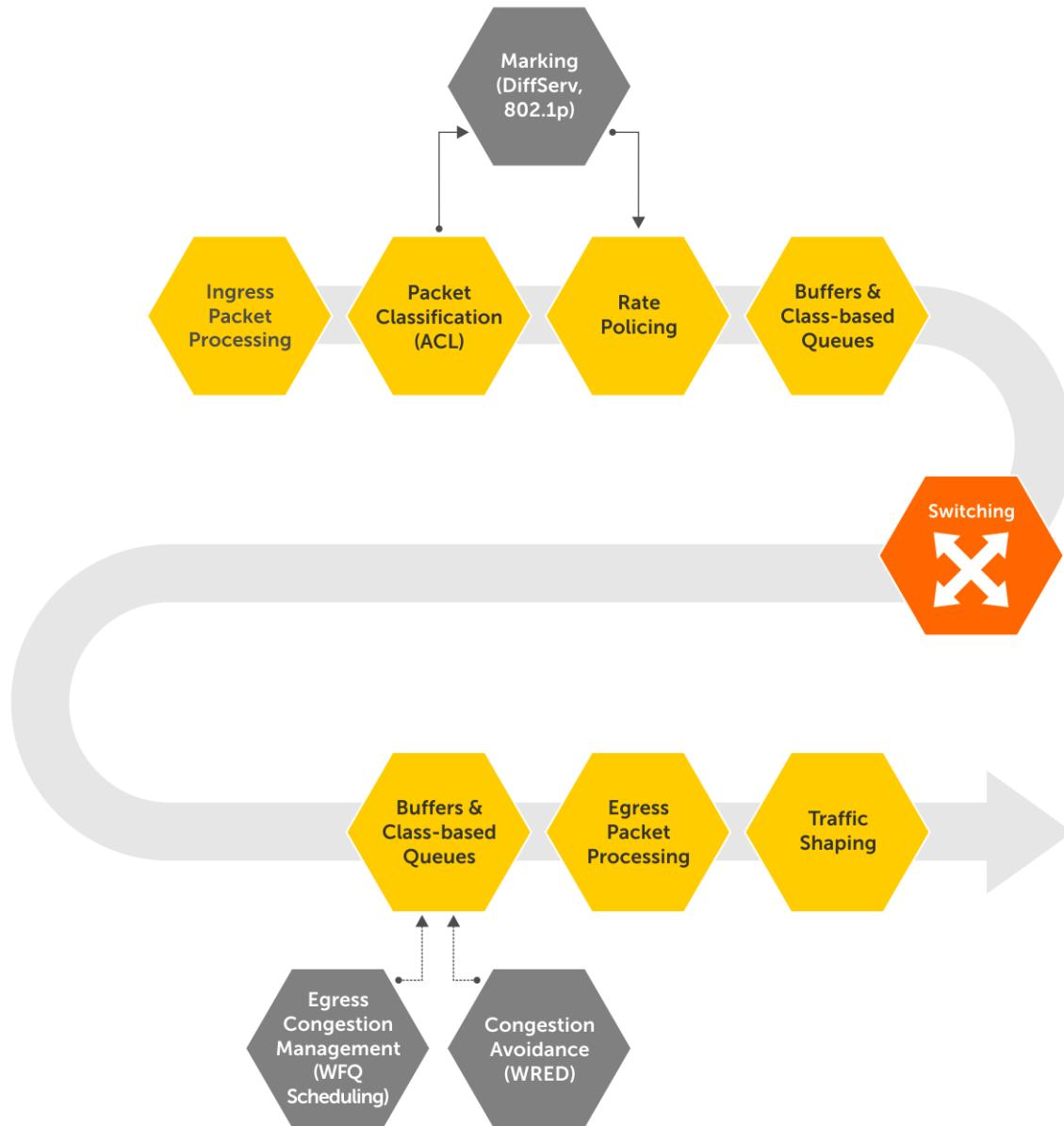
```
sonic# clear counters service-policy interface Eth1/1
```

```
sonic# clear counters service-policy policy-map qos_policy_map
```

# Quality of Service

Quality of Service (QoS) reserves network resources for highly critical application traffic with precedence over less critical application traffic. QoS prioritizes different types of traffic and ensures quality of service.

You can control these traffic flow parameters: delay, bandwidth, jitter, and drop. Different QoS features control the traffic flow parameters, as the traffic traverses a network device from ingress to egress interfaces.



## Topics:

- Flow-based QoS
- WRED and ECN
- Scheduler policy
- Port shaping
- QoS maps

- View QoS configuration
- Control plane policing
- Priority flow control
- Storm control
- Buffer management

## Flow-based QoS

Using classifiers that select traffic based on modular ACLs and L2-L4 header fields, you can create QoS policies that apply QoS actions, such as PCP or DSCP remarking and rate limiting, on matching traffic.

1. Use modular ACLs or L2-L4 header fields in class maps to select traffic in a flow.
2. In a policy map, configure the QoS actions to apply on classified traffic.
3. Apply the QoS policy on ingress or egress interfaces — globally all switch interfaces, a specified interface, the control plane, a VLAN, or a port channel.

## Classify traffic using modular ACLs

To select traffic for QoS policies, use an ACL or L2-L4 header classifiers.

### Classify traffic using modular ACLs

To classify traffic using modular ACLs:

1. Create an L2, IPv4, or IPv6 ACL to identify a traffic flow.

```
sonic(config) # {mac | ip | ipv6} access-list name
```

2. Add permit and deny rules to the ACL for L2 MAC, IPv4, or IPv6 traffic — see [Configure ACLs](#); for example:

```
Create IP ACL
sonic(config) # ip access-list pbr_v4_acl
sonic(conf-ipv4-acl) # seq 1 permit ip any 89.0.0.0/24 remark RULE_1
sonic(conf-ipv4-acl) # seq 2 permit ip any 89.0.1.0/24 remark RULE_2
sonic(conf-ipv4-acl) # seq 3 permit ip any 89.0.2.0/24 remark RULE_3
sonic(conf-ipv4-acl) # seq 4 permit ip any 89.0.3.0/24 remark RULE_4
sonic(conf-ipv4-acl) # seq 5 permit ip any 89.0.4.0/24 remark RULE_5
```

3. Create a classifier (class map) of match-type acl.

```
sonic(config) # class-map name match-type acl
```

Add the L2, IPv4, or IPv6 ACL to the class map to select flow traffic. Each class map uses only one ACL: L2 MAC, IPv4, or IPv6; for example:

```
Create class map for PBR IPv4 traffic
sonic(config) # class-map pbr_v4_class match-type acl
sonic(conf-class-map) # match access-group ip pbr_v4_acl
```

**(i) NOTE:** For a class map to be considered active, the ACL must be already configured. If the ACL is not configured, the classifier is incomplete and inactive. The class-map configuration is saved, and no error is displayed. When you configure the ACL, the classifier becomes active and applies any actions configured a policy.

### Classify traffic using L2-L4 header fields

For more fine-grained traffic classification in a flow, use match statements on L2, L3, and L4 header field values. You can combine match criteria for fields in different headers. For example, you can specify source MAC Address, VLAN, destination IP address, and TCP flags to identify a flow for forwarding actions. ACLs do not support this level of detailed packet classification. A class map is considered invalid if you configure mutually exclusive header fields, such as an IPv4 and an IPv6 address, as match criteria. If you enter no L2-L4 match statements in a class map, the classifier matches any traffic by default.

To classify traffic using L2-L4 header fields:

1. Create a classifier (class map) of match-type fields match-all.

```
sonic(config) # class-map name match-type fields match-all
```

- Add match statements to select packets based on L2, L3, and L4 header values — see [Configure flow-based ACLs](#); for example:

```
sonic(config)# class-map pbr_classmap match-type fields match-all
sonic(config-class-map)# match vlan 1001
sonic(config-class-map)# match destination-address mac host 00:01:00:11:00:11
sonic(config-class-map)# match destination-address ip 1.1.1.0/24
sonic(config-class-map)# match ip protocol tcp
sonic(config-class-map)# match tcp-flags syn rst
```

## Configure and apply QoS policies

A QoS policy specifies the actions to apply on a flow identified by a class map. QoS policies support the following actions:

- DSCP remarking — Sets the DSCP value in IPv4 and IPv6 headers to a specified value in matching packets.
- PCP remarking — Sets the Priority Code Point (PCP) value in 802.1Q VLAN headers to a specified value in matching packets.
- Policing — Applies rate-limiting settings to matching packets.
- Change traffic class — Sets a different traffic class/queue to prioritize or deprioritize matching traffic.

A policy can have multiple sections. Each section consists of a class map and associated actions in set statements. Also, each class map has a priority that indicates the order in which the classified traffic and actions are applied in the QoS policy; for example:

```
Create policy for QoS actions
sonic(config)# policy policy0 type qos

Configure QoS actions on classified traffic
sonic(conf-policy-map)# class class1 priority 100
sonic(conf-policy-map-flow)# police cir 10mbps cbs 20MB pir 50mbps pbs 100MB

Configure QoS actions on classified traffic
sonic(conf-policy-map)# class class0 priority 200
sonic(conf-policy-map-flow)# set pcp 5
sonic(conf-policy-map-flow)# set dscp 15

Apply QoS policy on interface
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# service-policy type qos in policy0
```

### QoS policy configuration

- Create an ACL class map to select traffic in flows using match statements in a class map — see [Configure flow-based ACLs](#).
- Create a QoS policy map to configure the actions to take on classified traffic. The policy-map name must begin with an alphanumeric character; 63 characters maximum. It can contain alphanumeric, hyphen (-), and underscore (\_) characters.

```
sonic(config)# policy-map name type qos
```

To delete the policy map, enter the `no policy-map name` command.

- Add a policy-map description (up to 256 characters).

```
sonic(conf-policy-map)# description text
```

- In policy-map configuration mode, add a class-map flow to the policy and enter policy-map flow configuration mode. Enter a priority number (0 to 4095) to specify the order in which a class map and its associated QoS actions are applied in the policy map. A higher priority class map is processed before a lower priority.

```
sonic(conf-policy-map)# class class-map-name priority number
sonic(conf-policy-map-flow) #
```

- In policy-map-flow configuration mode, add any of the following QoS actions to take on classified traffic.
  - Set the DSCP value in IP packet headers by entering a decimal value (0 to 63). For example, `set dscp 10` resets the six most significant bits of the DiffServ field to 001 010 for low drop probability. The default DSCP value is 0 (000 000) for best effort. For descriptions of other DSCP values, see RFC 2475.

```
sonic(conf-policy-map-flow) # set dscp dscp-value
```

To remove the configured DSCP remarking value, enter the `no set dscp` command.

- Set the PCP priority value in Ethernet frame headers (0 to 7).

```
sonic(conf-policy-map-flow) # set pcp pcp-value
```

To remove the configured DSCP remarking value, enter the `no set pcp pcp-value` command.

- Set a policing action for the matching class-map traffic. Configure the mandatory committed information rate (CIR) value. The CIR value specifies the data rate (desired bandwidth) for the classified traffic. Then configure a combination of optional policing values: committed burst size (CBS), peak information rate (PIR), and/or peak burst size (PBS).

**i NOTE:**

- If you configure only the CIR rate, a single-rate two-color policer is applied to classified traffic. Any traffic that exceeds the CIR value is marked as red and is dropped.
- If you configure both the CIR and PIR rates, a dual-rate three-color policer is applied to classified traffic. Any traffic that exceeds the CIR value, but is less than the PIR value, is marked as yellow and is not dropped. Any traffic that exceeds the PIR value is marked as red and is dropped.

```
sonic(conf-policy-map-flow) # police cir cir-value [kbps | mbps | gbps | tbps] [cbs cbs-value [KB | MB | GB | TB]] [pir pir-value [kbps | mbps | gbps | tbps]] [pbs pbs-value]
```

- `cir cir-value` — Enter the CIR value in bits per second; from 1 to 4 tbps (4000 gbps); there is no default. The CIR value specifies the amount of guaranteed bandwidth (without loss) for classified traffic. You can optionally enter a number followed by one of these suffixes to specify the `bps` rate: kbps (1000 bits per second), mbps (1,000,000 bits per second), gbps (1,000,000,000 bits per second), or tbps (1,000,000,000,000 bits per second).
- (Optional) `cbs cbs-value` — Enter the CBS value in bytes per second; from 1 to 500 GB or maximum value supported by ASIC; default is 20% more than the CIR value. The CBS value specifies the amount that the CIR value can be exceeded by traffic bursts and still be guaranteed. You can optionally enter a number followed by one of these suffixes to specify the `cbs-value`: KB (1000 bytes per second), MB (1,000,000 bytes per second), GB (1,000,000,000 bytes per second), or TB (1,000,000,000,000 bytes per second).
- (Optional) `pir pir-value` — Enter the PIR value in bits per second; from 1 to 4000 gbps (4 tbps); there is no default. The PIR value specifies the maximum amount of bandwidth allowed during normal traffic times without guarantee. The PIR value is greater than the CIR value. You can optionally enter a number followed by one of these suffixes to specify the `pir-value` rate: kbps, mbps, gbps, or tbps. The configured PIR value must be greater than the CIR value.
- (Optional) `pbs pbs-value` — Enter the PBS value in bytes per second; from 1 to 20% of the PIR value. The PBS value specifies the maximum amount of bandwidth allowed for traffic bursts without guarantee. The PBS value must be greater (in bytes) than the PIR and CIR values. The default PBS value is 20% of the configured PIR value in bytes. You can optionally enter a number followed by one of these suffixes to specify the `pbs-value`: KB, MB, GB , or TB.

To remove a configured policing action, enter the `no police [cir] [cbs] [pir] [pbs]` command.

- Reset the traffic class in the type-of-service (ToS) field in IP headers (0 to 7, from the least to greatest priority). For a description of the IP precedence (traffic class) values, see RFC 791.

```
sonic(conf-policy-map-flow) # set traffic-class precedence-value
```

To remove the configured traffic-class marking, enter the `no set traffic-class` command.

- Add additional class-map configurations with QoS actions to the policy map by repeating Steps 3 and 4.
- Apply a QoS policy map globally on all switch interfaces, on a specified interface, the control plane, a VLAN, or a port channel. To remove a policy from an interface, enter the `no` version of the command. You can apply QoS policies on ingress or egress interfaces.

- Globally on all switch interfaces:

```
sonic(config)# service-policy type qos {in | out} policy-map-name
```

- On an interface or subinterface:

```
sonic(conf-if-Eth) # service-policy type qos {in | out} policy-map-name
```

```
sonic(conf-subintf-Eth) # service-policy type qos {in | out} policy-map-name
```

- On the control plane:

```
sonic(conf-line-vty) # service-policy type qos in policy-map-name
```

- On VLAN interfaces:

```
sonic(config) # interface Vlan vlan-id
sonic(conf-if-Vlan) # service-policy type qos in policy-map-name
```

- On port-channel interfaces:

```
sonic(config) # interface PortChannel portchannel-number
sonic(conf-if-po) # service-policy type qos in policy-map-name
```

## Example: Flow-based QoS

This example shows how to use MAC and IP ACLs in class-maps to select L2 and L3 traffic in a flow and apply a QoS policy map with the actions to apply on the selected traffic.

```
Create L2 MAC ACL
sonic(config) # mac access-list 12_ACL_0
sonic(conf-mac-acl) # seq 1 permit any any ip
sonic(conf-mac-acl) # seq 2 permit any any ipv6
sonic(conf-mac-acl) # exit

Create IP ACL
sonic(config) # ip access-list 13_ACL_0
sonic(conf-ipv4-acl) # seq 1 permit tcp any any
sonic(conf-ipv4-acl) # seq 2 permit udp any any
sonic(conf-ipv4-acl) # exit

Create class map for IPv4 traffic
sonic(config) # class-map class0 match-type acl
sonic(config-class-map) # match access-group ip 13_ACL_0
sonic(config-class-map) # exit

Create class map for L2 MAC traffic
sonic(config) # class-map class1 match-type acl
sonic(config-class-map) # match access-group mac 12_ACL_0
sonic(config-class-map) # exit

Create policy for QoS actions
sonic(config) # policy-map policy0 type qos

Add class map and set actions for flow
sonic(config-policy-map) # class class0 priority 200
sonic(config-policy-map-flow) # set pcp 5
sonic(config-policy-map-flow) # set dscp 15
sonic(config-policy-map-flow) # exit

Add class map and police action for flow
sonic(config-policy-map) # class class1 priority 100
sonic(config-policy-map-flow) # police cir 10mbps cbs 20MB pir 50mbps pbs 100MB
sonic(config-policy-map-flow) # exit
sonic(config-policy-map) # exit

Apply policy to interfaces
SONiC(config) # interface Eth1/2
SONiC(conf-if-Eth1/2) # service-policy type qos in policy0

Verify flow-based QoS configuration
sonic# show policy-map policy0
Policy policy0 Type qos
 Description:
 Flow class0 at priority 200
 Description:
 set-pcp 5
 set-dscp 15
 Flow class1 at priority 100
 Description:
```

```

police cir 10000000 cbs 20000000 pir 50000000 pbs 100000000
Applied to:
 Eth1/2 at Ingress

sonic# show class-map class0
Class-map class0 match-type acl
Description:
Match:
 ip access-group 13_ACL_0
Referenced in flows:
 policy policy0 at priority 200

sonic# show service-policy interface Eth1/2
Eth1/2
Policy policy0 type qos at ingress
Description:
Flow class0 at priority 200 (Active)
Description:
 set-pcp 5
 set-dscp 15
 Packet matches: 0 frames 0 bytes
Flow class1 at priority 100 (Active)
Description:
 police: cir 10000000 cbs 20000000 pir 50000000 pbs 100000000 (Active)
 type bytes mode color-blind
 operational cir 0 cbs 0 pir 0 pbs 0
 green 0 packets 0 bytes action forward
 yellow 0 packets 0 bytes action forward
 red 0 packets 0 bytes action drop
 Packet matches: 0 frames 0 bytes

```

## WRED and ECN

### Weighted Random Detection (WRED)

Weighted Random Detection is a congestion avoidance mechanism that drops packets to prevent buffering resources from being consumed.

Network traffic is a mixture of packets and the rate at which some packets arrive is greater than others. Buffer space and traffic manager (BTM) — ingress or egress — space is consumed by only one or a few types of traffic, leaving no space for other types. By configuring WRED threshold values, you prevent specified traffic from consuming too much of the BTM resources.

Configure WRED parameters for a queue and specify the minimum and maximum thresholds, and drop rate. The minimum threshold is the allocated buffer space for specified traffic; for example, 1000KB on egress. If the 1000KB is consumed, packets are dropped randomly at an exponential rate until the maximum threshold is reached — this behavior is the “early detection” part of WRED.

If the maximum threshold (for example 2000KB) is reached, all incoming packets are dropped until specified traffic consumes less than 2000KB of the buffer space.

### Explicit Congestion Notification (ECN)

Explicit Congestion Notification enhances WRED by marking packets when the threshold value is exceeded, instead of dropping them.

Instead of dropping packets when the average queue length exceeds the minimum threshold value, ECN marks the Congestion Experienced (CE) bit of the ECN field in a packet as ECN-capable traffic (ECT).

### Configure WRED and ECN

1. Create a WRED policy and enter the policy name (up to 32 characters).

```
sonic(config)# qos wred-policy wred-policy-name
```

2. Set the bandwidth in KiloBytes (KB) per second allocated to the minimum and maximum threshold values, and the maximum drop rate for the green traffic class.

```
sonic(conf-wred-wred-green)# green minimum-threshold minimum-threshold-value maximum-threshold maximum-threshold-value drop-probability drop-probability-value
```

### 3. Configure ECN.

```
sonic(conf-wred-wred-green)# ecn [none | green]
```

- none — (Optional) Apply no filters
- green — (Optional) Filter only green traffic

### 4. Apply the WRED policy to an interface queue (0 to 7).

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# queue queue-number wred-policy-name
```

## View WRED policy

```
sonic# show qos wred-policy

Profile : test

Profile : wred-green
ecn : ecn_all
green-min-threshold : 100 KBytes
green-max-threshold : 200 KBytes
green-drop-probability : 50
```

## View WRED ECN counters

To view the WRED and ECN counters on all outbound queues of a specified interface, enter the `show queue wred-ecn counters interface` command.

```
sonic# show queue wred-ecn counters interface Ethslot/port[/breakout-port]
```

```
sonic# show queue wred-ecn counters interface Eth1/56

TxQ WRED Drops/Pkts ECN Marked/Pkts ECN Marked/Bytes

UC0 0 0 0 0
UC1 0 0 0 0
UC2 0 837922938 107207451904
UC3 390634 0 0 0
UC4 400000 0 0 0
UC5 0 0 0 0
UC6 0 0 0 0
UC7 0 0 0 0
UC8 0 0 0 0
UC9 0 0 0 0
```

To clear only the WRED and ECN counters on all queues of a specified interface, enter the `clear queue wred-ecn counters` command:

```
sonic# clear queue wred-ecn counters interface Ethslot/port[/breakout-port]
```

For example:

```
sonic# clear queue wred-ecn counters interface Eth1/56
sonic# show queue wred-ecn counters interface Eth1/56

TxQ WRED Drops/Pkts ECN Marked/Pkts ECN Marked/Bytes

UC0 0 0 0 0
UC1 0 0 0 0
UC2 0 0 0 0
UC3 0 0 0 0
UC4 0 0 0 0
UC5 0 0 0 0
UC6 0 0 0 0
UC7 0 0 0 0
UC8 0 0 0 0
UC9 0 0 0 0
```

# Scheduler policy

Create a QoS scheduler policy to configure queue parameters, such as scheduler algorithm, meter type, shaping, and bandwidth.

When you enable rate shaping, the system buffers all traffic that exceeds the specified rate until the buffer memory is exhausted. Rate shaping uses all buffers reserved for an interface or queue and shares buffer memory until it reaches the configured threshold.

You can allocate relative bandwidth (weight) to limit large flows and prioritize smaller flows. Allocate a relative amount of bandwidth to non-priority queues when priority queues are consuming maximum link bandwidth.

Schedule each egress queue of an interface using Weighted Deficit Round Robin (WDRR), Weighted Round Robin (WRR), or by strict-priority (SP), which are mutually exclusive.

## Configure scheduler policy

1. Create a scheduler policy and enter the policy name (up to 32 characters).

```
sonic(conf-sched-policy)# qos scheduler-policy scheduler-policy-name
```

2. Set the queue configuration. Enter the queue ID (0 to 7).

```
sonic(conf-sched-policy)# queue queue-id
```

3. Set the scheduler meter type to packets or bytes.

```
sonic(conf-sched-policy)# meter-type packets-or-bytes
```

4. Set the scheduler type: strict priority queuing, Weighted Round Robin (WRR), or Deficit Weighted Round Robin (DWRR).

```
sonic(conf-sched-policy-queue-q1)# type {strict | dwrr | wrr}
```

5. Configure the committed information rate in kilobytes per second (Kbps) for the amount of traffic sent in one flow.

```
sonic(conf-sched-policy-queue-q1)# cir cir-value
```

6. Configure the committed burst size in bytes for the amount of traffic can be sent at one time.

```
sonic(conf-sched-policy-queue-q1)# cbs cbs-value
```

7. Configure the peak information rate in kilobytes per second (Kbps) for the maximum number of bytes that can be sent at one time.

```
sonic(conf-sched-policy-queue-q1)# pir pir-value
```

8. Configure the peak burst size in bytes for the maximum number of bytes that can be sent at one time.

```
sonic(conf-sched-policy-queue-q1)# pbs pbs-value
```

9. Apply the scheduler policy to an interface.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# scheduler-policy scheduler-policy-name
```

## View scheduler policy

```
sonic# show qos scheduler-policy
Queue: 0
 type: strict
 weight: 10
 cir: 10000 Kbps
 cbs: 100 Bytes
 pir: 200000 Kbps
 pbs: 200 Bytes
Port:
 pir: 100 Kbps
 pbs: 200 Bytes
```

# Port shaping

Configure traffic shaping in a scheduler policy to set the peak traffic flow supported on an outbound interface and to match the QoS policies applied to the interface.

## Configure port shaping policy

1. Create a scheduler policy and enter the policy name (up to 32 characters).

```
sonic(config)# qos scheduler-policy scheduler-policy-name
```

2. Assign the scheduler policy to port interfaces.

```
sonic(conf-sched-policy)# port
```

3. Configure the peak information rate in kilobytes per second (Kbps).

```
sonic(conf-sched-policy-queue-port)# pir pir-value
```

4. Configure the peak burst size in bytes.

```
sonic(conf-sched-policy-queue-port)# pbs pbs-value
```

5. Return to CONFIGURATION mode.

```
sonic(conf-sched-policy-queue-port)# end
sonic# config terminal
```

6. Apply the scheduler policy to an interface.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# scheduler-policy scheduler-policy-name
```

## View scheduler policy

```
sonic# show qos scheduler-policy
Queue: 0
 type: strict
 weight: 10
 cir: 10000 Kbps
 cbs: 100 Bytes
 pir: 200000 Kbps
 pbs: 200 Bytes
Port:
 pir: 100 Kbps
 pbs: 200 Bytes
```

## View interface QoS configuration

```
sonic# show qos interface Eth1/3
 scheduler policy: p
 dscp-tc-map: test
 dot1p-tc-map: test
 tc-queue-map: test
 tc-pg-map: test
 pfc-priority-queue-map: test
 pfc-asymmetric: off
 pfc-priority : 3,4
```

# QoS maps

Use Quality of Service (QoS) mapping to classify traffic and configure actions to take on matching packets, such as queue or priority group assignment, or modifying the L2 802.1p or L3 DCSP value.

## DSCP to traffic class-map

Follow these steps to configure DSCP to traffic class-map.

1. Create a DSCP class-map and enter the class-map name (up to 32 characters).

```
sonic(config)# qos map dscp-tc dscp-tc-name
```

2. Add an entry to the DSCP to traffic class-map. The traffic-class value is 0 to 7.

```
sonic(conf-qos-map)# dscp dscp-value {traffic-class traffic-class-value}
```

3. Return to CONFIGURATION mode.

```
sonic(conf-qos-map)# end
sonic# config terminal
```

4. Apply the policy to an interface.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# qos-map dscp-tc dscp-tc-name
```

### View DSCP to traffic-class map

```
sonic# show qos map dscp-tc
DSCP-TC-MAP: test

 DSCP TC

 0 0
 1 1

```

## Dot1p to traffic class-map

Follow these steps to configure dot1p to traffic class-map.

1. Create a map to associate a set of traffic class to dot1p. Enter the policy map name (up to 32 characters).

```
sonic(config)# qos map dot1p-tc dot1p-tc-name
```

2. Add an entry to the dot1p to traffic class-map. The traffic class value is 0 to 7.

```
sonic(conf-qos-map)# dot1p dot1p-value traffic-class traffic-class-value
```

3. Return to CONFIGURATION mode.

```
sonic(conf-qos-map)# end
sonic# config terminal
```

4. Apply the policy to an interface.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# qos-map dot1p-tc dot1p-tc-name
```

### View dot1p to traffic class map

```
sonic# show qos map dot1p-tc
DOT1P-TC-MAP: test

 DOT1P TC
```

|   |   |
|---|---|
| 0 | 0 |
| 1 | 1 |

## Traffic class to queue map

Follow these steps to configure traffic-class to queue map.

1. Create a map to associate a set of traffic class to queue.

```
sonic(config) # qos map tc-queue tc-queue-name
```

2. Add an entry to map traffic class to queue.

```
sonic(conf-qos-map) # traffic-class tc {dot1p dot1p-val | dscp dscp-val | priority-group pg | queue qid}
```

3. Return to CONFIGURATION mode.

```
sonic(conf-qos-map) # exit
sonic# config terminal
```

4. Apply the policy to an interface.

```
sonic(config) # interface Eth1/2
sonic(conf-if-Eth1/2) # qos-map tc-queue tc-queue-name
```

### View traffic class to queue map

```
sonic# show qos map tc-queue
TC-Q-MAP: test

TC Q

0 0
1 1

```

## Traffic class to priority group map

Follow these steps to configure traffic-class to priority-group map.

1. Create a map to associate a set of traffic class to priority group. Enter the priority group name (up to 32 characters).

```
sonic(config) # qos map tc-pg tc-pg-name
```

2. Add an entry to map traffic class to queue.

```
sonic(conf-qos-map) # traffic-class tc {dot1p dot1p-val | dscp dscp-val | priority-group pg | queue qid}
```

3. Return to CONFIGURATION mode.

```
sonic(conf-qos-map) # exit
sonic# config terminal
```

4. Apply the policy to an interface.

```
sonic(config) # interface Eth1/2
sonic(conf-if-Eth1/2) # qos-map tc-queue tc-queue-name
```

## **View traffic class to priority group map**

```
sonic# show qos map tc-pg
TC-PG-MAP: test

 TC PG

 0 0
 1 1

```

## **Traffic class to dot1p map**

Follow these steps to configure traffic-class to dot1p map.

1. Create a map to associate a set of traffic class to dot1p. Enter the policy-map name (up to 32 characters).

```
sonic(config)# qos map tc-dot1p tc-dot1p-name
```

2. Add an entry to map traffic class to queue.

```
sonic(conf-qos-map)# traffic-class tc {dot1p dot1p-val | dscp dscp-val | priority-
group pg | queue qid}
```

3. Return to CONFIGURATION mode.

```
sonic(conf-qos-map)# exit
sonic# config terminal
```

4. Apply the policy to an interface.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# qos-map tc-dot1p tc-dot1p-name
```

## **Traffic class to DSCP map**

Follow these steps to configure traffic-class to DSCP map.

1. Create a map to associate a set of traffic class to a differentiated services code point (DSCP). Enter the policy name (up to 32 characters).

```
sonic(config)# qos map tc-dscp tc-dscp-name
```

2. Add an entry to map traffic class to queue.

```
sonic(conf-qos-map)# traffic-class tc {dot1p dot1p-val | dscp dscp-val | priority-
group pg | queue qid}
```

3. Return to CONFIGURATION mode.

```
sonic(conf-qos-map)# exit
sonic# config terminal
```

4. Apply the policy to an interface.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# qos-map tc-dscp tc-dscp-name
```

## **View traffic class-map to DSCP configuration**

```
sonic# show qos map tc-dscp
Traffic-Class-DSCP-MAP: td2

 TC DSCP

```

```
0 16
1 63
```

## PFC priority to queue map

Follow these steps to configure PFC priority to queue map.

1. Create a map to associate a set of priority flow control (PFC) priorities to queues. Enter the priority queue name (up to 32 characters).

```
sonic(config)# qos map pfc-priority-queue pfc-priority-queue-name
```

2. Add a PFC priority to queue entry in the map. Enter the values from 0 to 7.

```
sonic(conf-qos-map)# pfc-priority dot1p {queue qid}
```

3. Return to CONFIGURATION mode.

```
sonic(conf-qos-map)# exit
sonic# config terminal
```

4. Apply the policy to an interface.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# qos-map pfc-priority-queue pfc-priority-queue-name
```

### View PFC priority to queue map

```
sonic# show qos map pfc-priority-queue
PFC-PRIORITY-Q-MAP: test

 PRIORITY Q

 0 0
 1 1

```

## View QoS configuration

### View queue counters

```
sonic# show queue counters interface Eth1/2

TxQ Counter/pkts Counter/bytes Rate/PPS Rate/BPS Drop/pkts Drop/bytes

UC0 0 0 0/s 0/s 0 0
UC1 0 0 0/s 0/s 0 0
UC2 0 0 0/s 0/s 0 0
UC3 0 0 0/s 0/s 0 0
UC4 0 0 0/s 0/s 0 0
UC5 0 0 0/s 0/s 0 0
UC6 0 0 0/s 0/s 0 0
UC7 0 0 0/s 0/s 0 0
UC8 0 0 0/s 0/s 0 0
UC9 14964 3471648 0/s 0/s 0 0
MC10 0 0 0/s 0/s 0 0
MC11 0 0 0/s 0/s 0 0
MC12 0 0 0/s 0/s 0 0
MC13 0 0 0/s 0/s 0 0
MC14 0 0 0/s 0/s 0 0
MC15 0 0 0/s 0/s 0 0
MC16 0 0 0/s 0/s 0 0
MC17 0 0 0/s 0/s 0 0
MC18 0 0 0/s 0/s 0 0
MC19 0 0 0/s 0/s 0 0
```

## **View queue watermark**

```
sonic# show queue watermark unicast interface Eth1/2
Egress queue watermark per unicast queue:

UC0 UC1 UC2 UC3 UC4 UC5 UC6 UC7

0 0 0 0 0 0 0 0
```

```
sonic# show queue watermark multicast interface Eth1/2
Egress queue watermark per multicast queue:

MC8 MC9 MC10 MC11 MC12 MC13 MC14 MC15

0 0 0 0 0 0 0 0
```

## **View priority-group statistics**

```
sonic# show priority-group watermark headroom interface Eth1/2
Ingress headroom watermark per PG:

PG0 PG1 PG2 PG3 PG4 PG5 PG6 PG7

0 0 0 0 0 0 0 0
```

## **View global buffer pool statistics**

```
sonic# show buffer-pool watermark

Pool Bytes (Total) Bytes (Multicast)

egress_lossless_pool 0 0
egress_lossy_pool 0 0
ingress_lossless_pool 0 0
```

## **View interface buffer pool statistics**

```
sonic# show buffer-pool watermark interface Ethernet 0

Pool Bytes (Total) Bytes (Unicast)

egress_lossy_pool 0 0
egress_lossless_pool 0 0
ingress_lossless_pool 508 0
```

## **View device statistics**

```
sonic# show device watermark
Utilization (Bytes) : 1524
Utilization (Percent) : 0
```

## **Clear QoS statistics**

```
sonic# clear queue counters [interface Eth slot/port[/breakout-port] | CPU] queue-id
```

## **Clear queue statistics**

```
sonic# clear queue [watermark | persistent-watermark] {unicast|multicast|CPU} [interface
Eth slot/port[/breakout-port]]
```

## **Clear priority-group statistics**

```
sonic# clear priority-group [watermark | persistent-watermark] {headroom | shared}
[interface Eth slot/port[/breakout-port]]
```

### **Clear global buffer pool statistics**

```
sonic# clear buffer-pool watermark multicast
```

```
sonic# clear buffer-pool watermark unicast
```

### **Clear interface buffer pool statistics**

```
sonic# clear buffer-pool watermark interface Ethernet 0 shared
```

```
sonic# clear buffer-pool watermark interface Ethernet 0 unicast
```

### **Clear device buffer statistics**

```
sonic# clear device watermark
```

## **Control plane policing**

 **NOTE:** Precision Time Protocol (PTP) trap configuration in the CoPP system policy is supported only in the Cloud Premium and Enterprise Premium bundles. It is not available in the Cloud Standard, Enterprise Standard, and Edge Standard bundles.

Control plane policing (CoPP) increases security on the system by protecting the route processor from unnecessary traffic and providing priority to important control plane and management traffic. CoPP uses a dedicated control-plane configuration to set rate-limiting for control plane packets.

If the rate of control packets towards the CPU is higher than the packet rate that the CPU can handle, CoPP selectively drops some of the control traffic so that the CPU can process high-priority control traffic. In this way, CoPP provides increased security for the CPU from unwanted traffic and DDoS attacks, such as ping floods or TCP SYN floods.

### **Default CoPP system policy**

By default, a switch creates a system-wide CoPP policy (`copp-system-policy`). The default CoPP policy is automatically applied to the control plane to configure trap IDs for traffic punted to CPU, CPU queue assignments, and policers to rate limit CPU traffic. You can reconfigure the default system CoPP settings. To view the default CoPP policy, enter the `show policy type copp` command.

The switch also creates a default scheduler CoPP policy (`copp-scheduler-policy`). The default scheduler policy is automatically applied to configure CPU queue scheduling weights and CPU queue shaper rates. You can reconfigure the default CoPP scheduler settings. To view the default CoPP scheduler policy, enter the `show qos scheduler-policy copp-scheduler-policy` command.

### **ACL-based CoPP**

To configure CoPP with a finer granularity on a subset of CPU traffic, use modular ACL configuration to identify and process traffic in a flow — see [ACLs for flow-based services](#). Use ACL-based CoPP policies to:

- Reduce the policer rate for matching CPU traffic.
- Reprioritize matching CPU traffic by reassigning it to a different CPU queue than the default CoPP-assigned queue.

ACL-based CoPP policies apply only to ingress traffic destined to the CPU, and only to inband traffic on the CPU interface — not to management port traffic.

### **ACL CoPP rate-limiting restriction**

An ACL CoPP policy cannot increase rate above the system CoPP protocol transmission rate. For example, if the default rate for an IP protocol is 10 Mbps, you can configure a CoPP policy that reduces the rate for matching traffic from a specified source IP (SIP) address to 5 Mbps. The CoPP policy cannot increase the rate to a value (for example, 15 Mbps) that is higher than the default protocol rate of 10 Mbps.

In addition, the configured rate is not guaranteed for matching traffic, and serves only as the maximum rate at which matching traffic can be transmitted. In this example, matching traffic is not guaranteed to have 5 Mbps reserved.

## [Configure the default system CoPP policy](#)

### **Before you start**

Before you reconfigure any of the default system CoPP policy settings, display the default CoPP policies, supported protocol traffic, and actions.

To view the default CoPP system policy, enter the `show policy type copp` command.

```
sonic# show policy type copp
Policy copp-system-policy Type copp
Flow copp-system-bgp
 Action copp-system-bgp
 trap-action trap
 trap-priority 5
 trap-queue 5
 police cir 10000 cbs 10000
 meter-type: packets
 mode: sr_tcm
 red-action: drop
Flow copp-system-arp
 Action copp-system-arp
 trap-action: trap
 trap-priority: 4
 trap-queue: 4
 police cir 6000 cbs 6000
 meter-type: packets
 mode: sr_tcm
 red-action: drop
```

To view the default CoPP class map names used to select CPU traffic, enter the `show class-map match-type copp` command.

```
sonic# show class-map match-type copp
Classifier copp-system-bgp match-type copp
 protocol bgp
 protocol bgpv6
Classifier copp-system-arp match-type copp
 protocol arp_req
 protocol arp_resp
 protocol neigh_discovery
```

To view the supported CoPP protocol traffic and actions, enter the `show copp protocols` and `show copp actions` commands.

```
sonic# show copp protocols
Classifier match-type copp protocols
 protocol bgp
 protocol bgpv6
 protocol arp_req
 protocol arp_resp
 protocol neigh_discovery
```

```
sonic# show copp actions
CoPP action group copp-system-bgp
 trap-action trap
 trap-priority 5
 trap-queue 5
 police cir 10000 cbs 10000
 meter-type: packets
 mode: sr_tcm
 red-action: drop
CoPP action group copp-system-arp
 trap-action: trap
 trap-priority: 4
 trap-queue: 4
 police cir 6000 cbs 6000
 meter-type: packets
 mode: sr_tcm
 red-action: drop
```

To view the default CoPP scheduler policy, enter the `show qos scheduler-policy copp-scheduler-policy` command.

```
sonic# show qos scheduler-policy copp-scheduler-policy
Scheduler Policy: copp-scheduler-policy
Queue: 0
 type: wrr
 weight: 1
 pir: 100 Pps
Queue: 1
 type: wrr
 weight: 1
 pir: 100 Pps
```

### Reconfigure the default system CoPP policy

Enterprise SONiC provides a system-wide CoPP policy. You can change the default system CoPP settings. For example, you can specify different queue assignment, policing, and priority handling for CPU-destined Neighbor Discovery packets and ARP packets. To view the default CoPP settings, use the `show policy type copp` command.

1. Create a custom CoPP class map to select specified CPU protocol traffic. The supported custom CoPP class-map names are `copp-user-bgp` and `copp-user-arp`.

```
sonic(config)# class-map class-map-name match-type copp
```

To delete a custom class map, enter the `no class-map class-map-name` command. You cannot delete system CoPP class maps, such as `copp-system-bgp` and `copp-system-arp`.

- Enter the protocols used to match CPU traffic. To view the supported CoPP protocols, enter the `show copp protocols` command. For example, to create a custom ARP class map:

```
sonic(config)# class-map copp-user-arp match-type copp
sonic(config-class-map)# match protocol arp_req
sonic(config-class-map)# match protocol arp_resp
```

To separate Neighbor Discovery protocol traffic from ARP, create another class map:

```
sonic(config)# class-map copp-user-nd match-type copp
sonic(config-class-map)# match protocol neigh_discovery
```

To delete a protocol match statement, enter the `no` version of the command; for example:

```
sonic(config)# class-map copp-user-arp match-type copp
sonic(config-class-map)# no match protocol arp_resp
```

2. Create a CoPP action group to configure the trap and police actions to perform on the classified traffic.

```
sonic(config)# copp-action action-group-name
```

For example, to create an ARP CoPP action group:

```
sonic(config)# copp-action copp-user-arp
sonic(config-action) #
```

To create a Neighbor Discovery CoPP action group:

```
sonic(config)# copp-action copp-user-nd
```

To delete a CoPP action group, enter the `no copp-action action-group-name` command. You cannot delete a system CoPP action group.

- (Optional) Enter a trap action to take on CPU traffic. The supported trap actions are: `drop`, `forward`, `copy`, `copy_cancel`, `trap`, `log`, `deny`, and `transit` (default trap).

```
sonic(config-action)# set trap-action trap-action
```

```
sonic(config-action)# set trap-action copy
```

- (Optional) Specify the priority in which trap actions in the action group are performed in case CPU protocol traffic matches more than one CoPP class map. Valid priority values are 0 to 1023 (default 1).

```
sonic(config-action)# set trap-priority priority
```

```
sonic(config-action)# set trap-priority 30
```

To reset a trap priority to its default value, enter the `no set trap-priority` command.

- (Optional) Configure the CPU queue to which matching traffic in the action group is assigned. Valid CPU queue numbers are 0 to 47 (default 0).

```
sonic(config-action)# set trap-queue queue-number
```

```
sonic(config-action)# set trap-queue 30
```

To reset a trap queue to the default queue 0, enter the `no set trap-queue` command. You cannot change queue assignments in the system action groups `copp-system-ptp` and `copp-system-drop`.

- (Optional) Configure the policer settings used to reduce the throughput rate of matching traffic in the action group. Configure the mandatory committed information rate (CIR) value. The CIR value specifies the data rate (desired bandwidth) for the classified traffic. Configure an optional policing value for committed burst size (CBS). If you do not enter a `police` value, no rate limit is applied to the classified traffic in the CoPP action group.

```
sonic(conf-policy-map-flow)# police cir cir-pps-value [cbs cbs-packet-value]
```

- `cir cir-pps-value` — Enter the CIR value in packets per second; from 1 to 100000 pps; there is no default. The CIR value specifies the amount of guaranteed bandwidth (without loss) for classified traffic.
- (Optional) `cbs cbs-packet-value` — Enter the CBS value in number of packets; from 1 to 100000; the default depends on the default CoPP policy that is mapped to this CoPP action.

For example:

```
sonic(config-action)# police cir 2000 cbs 2000
```

To remove a configured policing action, enter the `no police [cir]` command.

3. Reconfigure the default system CoPP policy by applying the new, custom CoPP policy map on the ingress CPU interface. The `copp-system-policy` is the only CPU policy supported on the switch and cannot be deleted, only overwritten.

```
sonic(config)# policy-map copp-system-policy type copp
sonic(config-policy-map)#
```

- Add the custom CoPP class map to the policy map. For example, you can use the custom CoPP class-map names `copp-user-arp` and `copp-user-nd` that you created in Step 1.

```
sonic(config-policy-map)# class-map class-map-name
sonic(config-policy-map-flow)#+
```

To delete the custom class map from the policy, enter the `no class class-map-name` command.

- Bind a custom CoPP action group to the class map in the CoPP policy.

```
sonic(config-policy-map)# set copp-action action-group-name
```

For example:

```
sonic(config)# policy-map copp-system-policy type copp
sonic(config-policy-map)# class copp-system-arp
sonic(config-policy-map-flow)# set copp-action copp-system-arp
```

**i** **NOTE:** Do not use the system CoPP class maps `copp-system-sflow` and `copp-system-ptp` in a CoPP policy map that contains trap action groups with other class maps. SFlow and PTP traps should be configured separately in their own CoPP action group.

## Reconfigure the default system CoPP scheduler policy

Enterprise SONiC provides a system-wide CoPP scheduler policy. You can change the default scheduler settings. For example, you can change the priority and peak rates of specified CPU queue traffic. To view the default scheduler settings, use the `show policy type copp` command.

1. Access the CoPP scheduler policy and specify the CPU queue to reconfigure.

```
sonic(config)# qos scheduler-policy copp-scheduler-policy
sonic(conf-sched-policy-copp-scheduler-policy)# queue number
sonic(conf-scheduler-copp-scheduler-policy-queue)#
```

2. Set the peak information rate (PIR) to shape CPU queue traffic. Whereas policing drops excess traffic that exceeds the configured maximum CIR and PIR rates, shaping stores excess traffic packets in a queue and reschedules them for later transmission. Enter the PIR value in packets per second (pps) from 1 to 100000; there is no default.

```
sonic(conf-scheduler-copp-scheduler-policy-queue)# pir pir-pps-value
```

For example:

```
sonic(config)# qos scheduler-policy copp-scheduler-policy
sonic(conf-sched-policy-copp-scheduler-policy)# queue 3
sonic(conf-scheduler-copp-scheduler-policy-queue-3)# pir 2000
```

To remove a configured policing action, enter the `no pir` command.

3. Set the default weight applied to CPU traffic in the specified queue. The weight value determines the priority with which CPU queue traffic is scheduled. Enter the weight value as a number from 1 to 1000 (default 1).

```
sonic(conf-scheduler-copp-scheduler-policy-queue)# weight number
```

For example:

```
sonic(config)# qos scheduler-policy copp-scheduler-policy
sonic(conf-sched-policy-copp-scheduler-policy)# queue 3
sonic(conf-scheduler-copp-scheduler-policy-queue-3)# weight 2
```

To remove a configured weight, enter the `no weight` command.

### **Example: Reconfigure CoPP and scheduler policy**

This example shows how to reduce the default police action from 6000 pps to 2000 pps in a custom CoPP action group IP2ME traffic, and reset the peak information rate of the CoPP scheduler policy for queue 7 — the queue to which the action group reassigns classified CPU traffic.

```
sonic# show copp actions | find copp-system-ip2me
CoPP action group copp-system-ip2me
 trap-action trap
 trap-priority 7
 trap-queue 7
 police cir 6000 cbs 6000
 meter-type packets
 mode sr_tcm
 red-action drop
sonic# configure terminal
sonic(config)# copp-action copp-system-ip2me
sonic(config-action)# police cir 2000 cbs 2000
sonic(config-action)# exit
sonic(config)# qos scheduler-policy copp-scheduler-policy
sonic(conf-sched-policy-copp-scheduler-policy)# queue 7
sonic(conf-scheduler-copp-scheduler-policy-queue-7)# pir 2000
sonic(conf-scheduler-copp-scheduler-policy-queue-7)# end
```

### **View CPU queue counters**

To verify that CPU traffic is handled by the desired queue and that the policer rate is honored, for CPU traffic types, use the `show queue counters interface CPU` command.

```
sonic# show queue counters interface CPU [queue queue-number]
```

```
sonic# sonic# show queue counters interface CPU queue 1

```

| TxQ | Counter/pkts | Rate/PPS | Rate/bPS | Drop/pkts | Drop/bytes |
|-----|--------------|----------|----------|-----------|------------|
| MC1 | 549712       | 7999/s   | 831970/s | 0         | 0          |

## Configure ACL CoPP policy

Use an ACL-based CoPP policy to change the queue assignment and policer rate of more finely selected CPU traffic.

1. Create a class map to select a CPU traffic flow. A class-map can use either a MAC, IP, or IPv6 ACL to match the incoming traffic or fields from L2, L3, and L4 headers.
  - To configure a MAC, IP, or IPv6 ACL, see [Configure ACLs](#).
  - To match the incoming traffic or fields from L2, L3, and L4 headers, see [Configure flow-based ACLs](#).
2. Create a policy map to configure the actions to take on classified CPU traffic. The policy-map name must begin with an alphanumeric character; 63 characters maximum. It can contain alphanumeric, hyphen (-), and underscore (\_) characters.

```
sonic(config)# policy-map name type acl-copp
```

To delete the policy map, enter the `no policy-map name` command.

- Add a policy-map description (up to 256 characters).

```
sonic(conf-policy-map)# description text
```

- Add a class-map flow to the CPU policy. Enter a priority number (0 to 4095) to specify the order in which a class map and its associated QoS actions are applied in the policy map. A higher priority class map is processed before a lower priority.

```
sonic(conf-policy-map)# class class-map-name priority number
sonic(conf-policy-map-flow) #
```

3. In policy-map-flow mode, add any of the following actions to take on the classified CPU traffic.

- Set the CPU queue to which matching traffic is assigned (0 to 31). To view the current queue assignments for CPU protocol traffic, use the `show policy type copp` command.

```
sonic(conf-policy-map-flow)# set trap-queue queue-number
```

To unconfigure the trap-queue setting and return to the default, enter the `no set trap-queue` command.

- Set a policing action for matching class-map traffic to reduce the throughput rate. Configure the mandatory committed information rate (CIR) value. The CIR value specifies the data rate (desired bandwidth) for the classified traffic. Then configure a combination of optional policing values: committed burst size (CBS), peak information rate (PIR), and/or peak burst size (PBS).

**i NOTE:**

- If you configure only the CIR rate, a single-rate two-color policer is applied to classified traffic. Any traffic that exceeds the CIR value is marked as red and is dropped.
- If you configure both the CIR and PIR rates, a dual-rate three-color policer is applied to classified traffic. Any traffic that exceeds the CIR value, but is less than the PIR value, is marked as yellow and is not dropped. Any traffic that exceeds the PIR value is marked as red and is dropped.

```
sonic(conf-policy-map-flow)# police cir cir-value [kbps | mbps | gbps | tbps] [cbs cbs-value [KB | MB | GB | TB]] [pir pir-value [kbps | mbps | gbps | tbps]] [pbs pbs-value]
```

- `cir cir-value` — Enter the CIR value in bits per second; from 1 to 4 Tbps (4000 Gbps); there is no default. The CIR value specifies the amount of guaranteed bandwidth (without loss) for classified traffic. You can optionally enter a number followed by one of these suffixes to specify the `bps` rate: kbps (1000 bits per second), mbps (1,000,000 bits per second), gbps (1,000,000,000 bits per second), or tbps (1,000,000,000,000 bits per second).
- (Optional) `cbs cbs-value` — Enter the CBS value in bytes per second; from 1 to 500 GB or the maximum value supported by ASIC; default is 20% more than the CIR value. The CBS value specifies the amount that the CIR value can be exceeded by traffic bursts and still be guaranteed. You can optionally enter a number followed by one of these suffixes to specify the `cbs-value`: KB (1000 bytes per second), MB (1,000,000 bytes per second), GB (1,000,000,000 bytes per second), or TB (1,000,000,000,000 bytes per second).

- (Optional) `pir pir-value` — Enter the PIR value in bits per second; from 1 to 4000 Gbps (4 Tbps); there is no default. The PIR value specifies the maximum amount of bandwidth allowed during normal traffic times without guarantee. The PIR value is greater than the CIR value. You can optionally enter a number followed by one of these suffixes to specify the `pir-value` rate: kbps, mbps, gbps, or tbps. The configured PIR value must be greater than the CIR value.
- (Optional) `pbs pbs-value` — Enter the PBS value in bytes per second; from 1 to 20% of the PIR value. The PBS value specifies the maximum amount of bandwidth allowed for traffic bursts without guarantee. The PBS value must be greater (in bytes) than the PIR and CIR values. The default PBS value is 20% of the configured PIR value in bytes. You can optionally enter a number followed by one of these suffixes to specify the `pbs-value`: KB, MB, GB , or TB.

To remove a configured policing action, enter the `no police [cir] [cbs] [pir] [pbs]` command.

4. Apply a policy map on the ingress CPU interface. To remove a policy from an interface, enter the `no` version of the command. You can only apply an ACL-CoPP policy on the CPU.

```
sonic(config)# interface CPU
sonic(conf-if-CPU)# service-policy type acl-copp in policy-map-name
```

#### Create CoPP policy

```
sonic(config)# policy-map policy_ip type acl-copp
sonic(config-policy-map)# class class_ip priority 100
sonic(config-policy-map-flow)# set trap-queue 1
sonic(config-policy-map-flow)# police cir 1024000 cbs 1024000
sonic(config-policy-map-flow)# exit
sonic(config-policy-map)# exit
sonic(config)# interface CPU
sonic(conf-if-CPU)# service-policy type acl-copp in policy_ip
```

#### View CoPP policy

```
sonic# show service-policy type acl-copp
```

```
sonic# show policy-map type acl-copp
Policy policy_ip Type acl-copp
Description:
Flow class_ip at priority 100
Description:
police cir 1024000 cbs 1024000 pir 0 pbs 0
set-trap-queue 1
Applied to:
CPU at Ingress
```

#### View CoPP counters and policy binding

```
sonic# show service-policy interface CPU
```

```
sonic# show service-policy interface CPU
CPU
Policy policy_ip type acl-copp at ingress
Description:
Flow class_ip at priority 100 (Active)
Description:
set-trap-queue 1
police: cir 1024000 cbs 1024000 pir 0 pbs 0 (Active)
type bytes mode color-blind
operational cir 512000 cbs 1024000 pir 512000 pbs 1024000
green 1805 packets 238260 bytes action forward
yellow 0 packets 0 bytes action forward
red 0 packets 0 bytes action drop
Packet matches: 152010246 frames 19457414272 bytes
```

# Priority flow control

In a converged data-center network, use priority flow control (PFC) to ensure that no frames are lost due to congestion. PFC uses the 802.1p priority in the Ethernet header to pause priority-specific traffic sent from a transmitting device. The 802.1p priority is also known as the class of service (CoS) or dot1p priority value.

When PFC detects congestion in a dot1p or DSCP traffic class, it sends a pause frame for the priority traffic to the transmitting device. In this way, PFC ensures that the switch does not drop specified priority traffic. PFC handles traffic congestion by pausing prioritized dot1p or DSCP traffic on an ingress interface while transmitting other dot1p or DSCP traffic as best-effort, also known as lossy data transmission.

PFC enhances the existing 802.3x pause capability to enable flow control based on 802.1p priorities. Instead of stopping all traffic on a link, as performed by the 802.3x pause mechanism, PFC pauses traffic for 802.1p traffic types. For example, when LAN traffic congestion occurs on an interface, PFC ensures lossless flows of storage and server traffic while allowing for lossy best-effort transmission of other traffic.

## PFC configuration notes

- Enterprise SONiC supports lossless RoCEv2 with default configurations, including WRED/ECN, scheduling, QoS maps, PFC, and PFC watchdog — see [Enable RoCEv2 with default configuration](#).
- Lossless operation is supported only for unicast traffic. The multicast traffic pause setting is not honored.
- Dell Technologies recommends you handle multicast traffic as follows: Either do not manage multicast traffic on queues 3 and 4 or use interface storm control to drop multicast packets if multicast is not required in a data center.

## Configure priority flow control

1. Enable the pre-configured switch-specific QoS buffer settings that PFC uses — see [Buffer management](#).

```
sonic(config) # buffer init lossless
```

2. Create a DSCP class map and configure the assignment of received DSCP values to a traffic class.

```
sonic(config) # qos map dscp-tc dscp-tc-name
sonic(conf-qos-map) # dscp dscp-value traffic-class traffic-class-value
```

For example:

```
sonic(config) # qos map dscp-tc pfc34
sonic(conf-qos-map) # dscp 3 traffic-class 3
sonic(conf-qos-map) # dscp 4 traffic-class 4
sonic(conf-qos-map) # exit
```

3. Create a QoS map to assign ingress traffic classes to a priority queue. The traffic class of incoming traffic is mapped to the specified transmit queue on an egress interface.

```
sonic(config) # qos map tc-queue tc-queue-name
sonic(conf-qos-map) # traffic-class traffic-class-value queue queue-value
```

For example:

```
sonic(config) # qos map tc-queue pfc34
sonic(conf-qos-map) # traffic-class 0 queue 0
sonic(conf-qos-map) # traffic-class 1 queue 1
sonic(conf-qos-map) # traffic-class 2 queue 2
sonic(conf-qos-map) # traffic-class 3 queue 3
sonic(conf-qos-map) # traffic-class 4 queue 4
sonic(conf-qos-map) # traffic-class 5 queue 5
sonic(conf-qos-map) # traffic-class 6 queue 6
sonic(conf-qos-map) # traffic-class 7 queue 7
sonic(conf-qos-map) # exit
```

4. Create a QoS map to assign ingress traffic classes to a PFC priority group.

**i** **NOTE:** For VLAN-tagged traffic, priority group assignment is made using a packet's dot1p value instead of the tc-pg mapping. Dell Technologies recommends that you map dot1p 3 traffic to traffic class 3, and map dot1p 4 traffic to traffic class 4.

```
sonic(config) # qos map tc-pg tc-pg-name
sonic(conf-qos-map) # traffic-class traffic-class-value priority-group priority-group-value
```

For example:

```
sonic(config) # qos map tc-pg pfc34
sonic(conf-qos-map) # traffic-class 0 priority-group 0
sonic(conf-qos-map) # traffic-class 1 priority-group 0
sonic(conf-qos-map) # traffic-class 2 priority-group 0
sonic(conf-qos-map) # traffic-class 3 priority-group 3
sonic(conf-qos-map) # traffic-class 4 priority-group 4
sonic(conf-qos-map) # traffic-class 5 priority-group 0
sonic(conf-qos-map) # traffic-class 6 priority-group 0
sonic(conf-qos-map) # traffic-class 7 priority-group 0
sonic(conf-qos-map) # exit
```

5. Create a QoS map to assign ingress PFC priority traffic to a queue. When a peer device sends PFC pause frames with a PFC priority, the switch stops sending traffic from the corresponding queue.

```
sonic(config) # qos map pfc-priority-queue pfc-priority-queue-name
sonic(conf-qos-map) # pfc-priority pfc-priority-value queue queue-value
```

For example:

```
sonic(config) # qos map pfc-priority-queue pfc34
sonic(conf-qos-map) # pfc-priority 0 queue 0
sonic(conf-qos-map) # pfc-priority 1 queue 1
sonic(conf-qos-map) # pfc-priority 2 queue 2
sonic(conf-qos-map) # pfc-priority 3 queue 3
sonic(conf-qos-map) # pfc-priority 4 queue 4
sonic(conf-qos-map) # pfc-priority 5 queue 5
sonic(conf-qos-map) # pfc-priority 6 queue 6
sonic(conf-qos-map) # pfc-priority 7 queue 7
sonic(conf-qos-map) # exit
```

6. Apply the configured QoS maps on interfaces to configure DSCP-based traffic classification and traffic class-based queue assignment. Enable PFC for specified PFC priority values.

**i** **NOTE:** PFC supports only PFC priority values 3 and 4.

```
sonic(config) # interface Eth slot/port[/breakout-port]
sonic(conf-if-Eth) # qos-map dscp-tc dscp-tc-name
sonic(conf-if-Eth) # qos-map tc-queue tc-queue-name
sonic(conf-if-Eth) # qos-map tc-pg tc-pg-name
sonic(conf-if-Eth) # qos-map pfc-priority-queue pfc-priority-queue-name
sonic(conf-if-Eth) # priority-flow-control priority pfc-priority-value
```

For example:

```
sonic(config) # interface Eth slot/port[/breakout-port]
sonic(conf-if-Eth) # qos-map dscp-tc pfc34
sonic(conf-if-Eth) # qos-map tc-queue pfc34
sonic(conf-if-Eth) # qos-map tc-pg pfc34
sonic(conf-if-Eth) # qos-map pfc-priority-queue pfc34
sonic(conf-if-Eth) # priority-flow-control priority 3
sonic(conf-if-Eth) # priority-flow-control priority 4
```

## Configure asymmetric PFC

When PFC is enabled, an interface sends PFC pause frames to stop a peer from sending lossless priority traffic when a lossless queue is congested. You can also enable asymmetric PFC so that when PFC pause frames are received from a peer, the interface honors pause frames on all PFC priorities, not only lossless priority traffic, and stops sending packets. The interface continues to send pause frames in case of lossless queue congestion.

Asymmetric PFC is disabled by default. Enable asymmetric PFC on a per-interface basis by entering the `priority-flow-control asymmetric` command; for example:

```
sonic(config)# interface Eth slot/port[/breakout-port]
sonic(conf-if-Eth)# priority-flow-control asymmetric
```

To disable asymmetric PFC, enter the `no priority-flow-control asymmetric` command. To display the configured asymmetric PFC mode on an interface, enter the `show interface status` command.

```
sonic# show interface status
 Interface Lanes Speed MTU Alias Oper Admin Type Asym PFC
----- -----
Eth1/2 0,1,2,3 40G 9100 etp1 up up QSFP+ on
Eth1/3 4,5,6,7 40G 9100 etp2 up up QSFP+ off
Eth1/4 8,9,10,11 40G 9100 etp3 up up QSFP+ off
Eth1/5 12,13,14,15 40G 9100 etp4 up up QSFP+ off
```

### Configure PFC watchdog

The PFC watchdog detects and mitigates a PFC storm received on a port. PFC pause frames are used for lossless Ethernet priority traffic to pause a peer switch from sending packets. This back-pressure mechanism can propagate to the whole network and cause the network to stop forwarding traffic. PFC watchdog detects abnormal back-pressure caused by receiving an excessive amount of PFC pause frames, and mitigates the situation by temporarily disabling PFC pause operation. PFC watchdog performs three functions:

- PFC storm detection — Detects that a lossless queue is receiving a PFC storm from a peer switch and that the queue is in a paused state for the configured detection time. You enable PFC storm detection on a per-port basis. It is supported only on lossless queues. By default, PFC storm detection is disabled.
- PFC storm mitigation — After a PFC storm is detected on a queue, PFC watchdog can drop and forward traffic on a per-queue basis. The default action is drop. When the drop action is enabled:
  - All existing packets in the output queue are dropped.
  - All packets destined to the output queue are dropped.
  - All packets received for the priority group on the queue are dropped, including received pause frames. As a result, the switch does not send pause frames to its neighbor due to congestion in the output queue.If you enable the forward action, the queue no longer honors the PFC frames that it receives. All packets destined to the queue are forwarded as well as the packets that are in the queue.
- PFC storm restoration — The PFC watchdog monitors PFC frames received on a lossless queue. When no PFC frames are received for the configured restore time, PFC is re-enabled on the queue. If PFC storm mitigation is set to drop, PFC watchdog stops dropping packets. Configure the restoration time on a per-port level.

1. Globally enable the collection of PFC watchdog counters on all interlaces, including incoming pause frames, queue counters, and ACL counters.

```
sonic(config)# priority-flow-control watchdog counter-poll
```

2. Configure the PFC watchdog polling interval for checking incoming packets. Specify an interval by entering a number in milliseconds, from 100 to 3000.

```
sonic(config)# priority-flow-control watchdog polling-interval milliseconds
```

3. On a per-interface basis:
  - Configure the restore time for monitoring the PFC frames received on a lossless queue; in milliseconds, from 100 to 60000. When no PFC frames are received for the configured restore time, PFC is re-enabled on the queue.
  - Configure the PFC watchdog action to take after a PFC storm is detected on a lossless queue:
    - alert — Sends an alert message when a PFC storm is detected.
    - drop — Drops incoming packets destined for lossless queues.
    - forward — Forwards all packets destined to the queue and all packets in the queue.
  - Configure the detection time for detecting a PFC storm from a peer switch on lossless queues (in milliseconds, from 100 to 5000) and enable PFC watchdog on the interface.

```
sonic(conf-if-Eth)# priority-flow-control watchdog restore-time milliseconds
sonic(conf-if-Eth)# priority-flow-control watchdog action {alert| drop | forward}
sonic(conf-if-Eth)# priority-flow-control watchdog on detect-time milliseconds
```

To disable PFC watchdog on an interface, enter the `priority-flow-control watchdog off` command.

## Example: Configure PFC watchdog

```
sonic(config)# priority-flow-control watchdog counter-poll
sonic(config)# priority-flow-control watchdog polling-interval 1000
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# priority-flow-control watchdog restore-time 200
sonic(conf-if-Eth1/2)# priority-flow-control watchdog action drop
sonic(conf-if-Eth1/2)# priority-flow-control watchdog on detect-time 100
```

## View PFC configuration

```
sonic(conf-if-Eth1/2)# show configuration
!
interface Eth1/2
mtu 9100
speed 100000
fec none
shutdown
qos-map dscp-tc pfc34
qos-map tc-queue pfc34
qos-map tc-pg pfc34
qos-map pfc-priority-queue pfc34
priority-flow-control priority 3
priority-flow-control priority 4
priority-flow-control asymmetric
priority-flow-control watchdog action drop
priority-flow-control watchdog on detect-time 100
priority-flow-control watchdog restore-time 200
```

```
sonic# show qos interface Eth1/2
dscp-tc-map: pfc34
tc-queue-map: pfc34
tc-pg-map: pfc34
pfc-priority-queue-map: pfc34
pfc-asymmetric: on
pfc-priority : 3,4
PFC Watchdog
 Status : off
 Action : drop
 Detection Time : 100ms
 Restoration Time : 200ms
```

```
sonic# show qos interface Eth all
 | Interface Maps | Priority-Flow-Control
 |Scheduler| dscp- dot1p- fg- fg- fg- fg- pfc- |asym ----- WATCHDOG -----
Interface |Policy | fg fg queue pg dscp dot1p p2q |mode priority action detect restore
-----+-----+-----+-----+-----+-----+-----+-----+
Eth1/2 |scheduler.0| pfc34 pfc34 pfc34 pfc34| on 3,4 DROP 100 200
Eth1/3 |scheduler.1| AZURE AZURE AZURE AZURE| off
Eth1/4 |scheduler.0| AZURE AZURE AZURE AZURE| off
```

## View PFC statistics by PFC priority

```
sonic# show qos interface Eth1/2 priority-flow-control statistics
Flow Control frames received
Interface PFC0 PFC1 PFC2 PFC3 PFC4 PFC5 PFC6 PFC7
----- ---- ---- ---- ---- ---- ---- ----
Eth1/2 0 0 0 0 0 0 0 0

Flow Control frames transmitted
Interface PFC0 PFC1 PFC2 PFC3 PFC4 PFC5 PFC6 PFC7
----- ---- ---- ---- ---- ---- ---- ----
Eth1/2 0 0 0 0 0 0 0 0
```

## View PFC watchdog configuration

**NOTE:** Flex counters come from the mechanism used by PFC watchdog at the configured polling interval. Flex counters are enabled with the `priority-flow-control watchdog counter-poll` command.

```
sonic# show priority-flow-control watchdog

Watchdog Summary

Polling Interval: : 100
Flex Counters: : enabled
```

### View PFC watchdog statistics

```
sonic# show qos interface Eth1/2 queue 3 priority-flow-control statistics
```

#### PFC Watchdog Statistics

| Interface | Queue | Status | Storms   |          | Transmitted |      | Received |      | TX |      | RX |      |
|-----------|-------|--------|----------|----------|-------------|------|----------|------|----|------|----|------|
|           |       |        | Detected | Restored | OK          | Drop | OK       | Drop | OK | Drop | OK | Drop |
| Eth1/2    | 0     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 1     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 2     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 3     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 4     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 5     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 6     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 7     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 8     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 9     | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 10    | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 11    | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |
| Eth1/2    | 12    | N/A    | 0        | 0        | 0           | 0    | 0        | 0    | 0  | 0    | 0  | 0    |

## Storm control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. The type of traffic can be broadcast, unknown-unicast, or unknown-multicast (BUM).

The storm-control feature allows the user to limit the amount of BUM traffic admitted to the system. This can be achieved by configuring the type of storm (broadcast or unknown-unicast or unknown-multicast), and the corresponding kilo bits per second (kbps) parameter on a given physical interface. Traffic that exceeds the configured rate is dropped.

Unknown-multicast traffic consists of all multicast traffic which does not match any of the statically configured or dynamically learned multicast groups.

## Buffer management

On an Enterprise SONiC switch, the memory buffer is divided into an ingress and egress buffer pool with fixed amounts of allocated memory. An incoming packet must be admitted to both the ingress and egress buffer pools to be transmitted over switch ports.

On a port interface, QoS priority groups manage ingress traffic admission to the memory management unit (MMU); queues manage egress traffic admission to the MMU.

- Each priority group and queue receives a buffer from available buffer pools using the settings in the assigned buffer profile — see [Configure buffer pools](#).
- To map DSCP and dot1p traffic classes to priority groups and port queues, see [Quality of Service](#).

### Buffer management notes

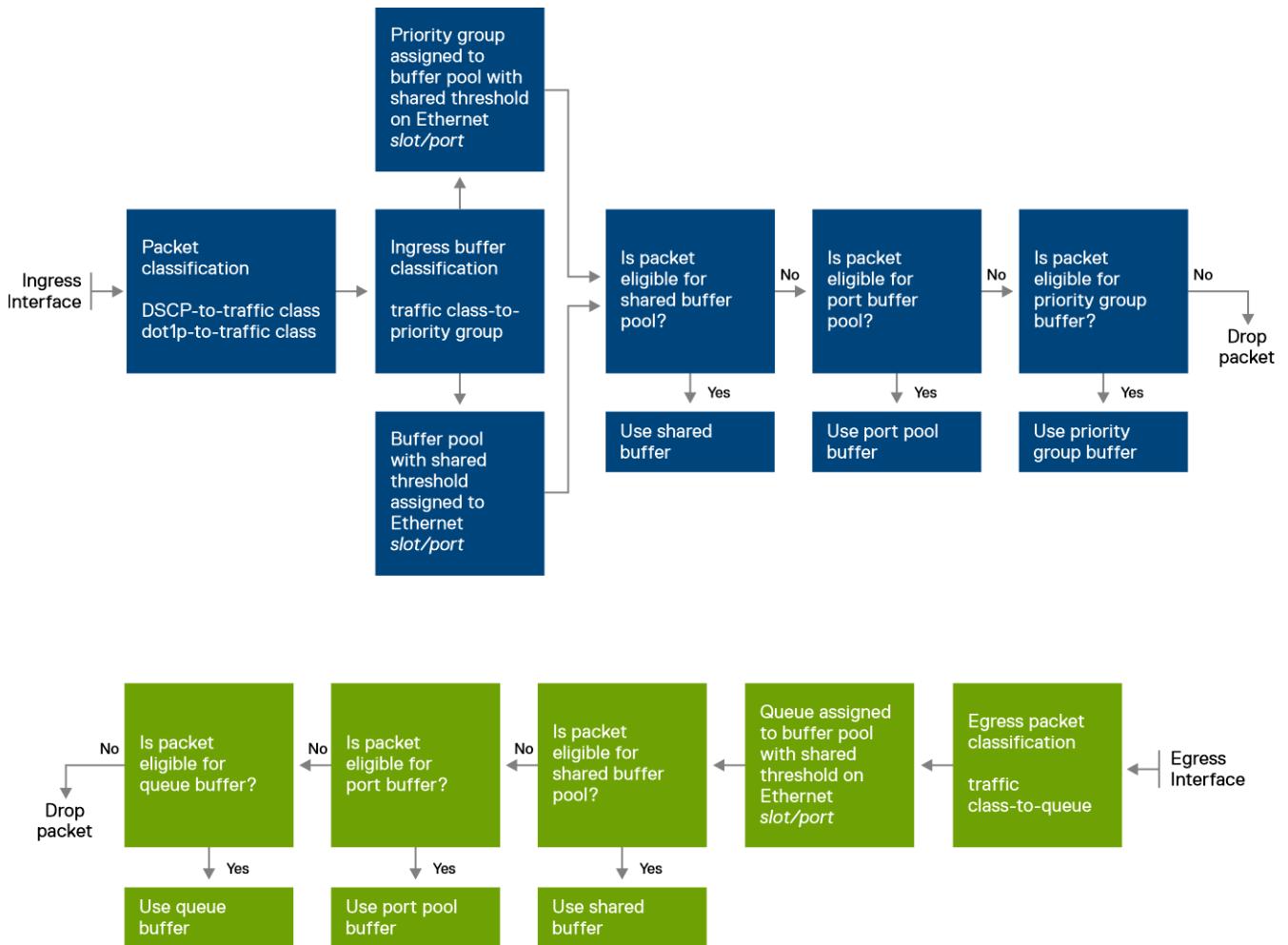
- Buffer pools can have shared and reserved memory.
  - A reserved buffer is a fixed amount of allocated memory that you configure for priority groups/queues as guaranteed buffers, such as the minimum egress queue or minimum ingress priority-group buffering allowed for ingress or egress traffic, or the shared headroom pool (xoff) in the ingress lossless pool. The total amount of available buffer pool memory is platform-specific.

**(i) NOTE:** The shared headroom size is the available shared buffer that can be used by all lossless priority groups after generating a pause/Xoff (transmit off) packet. This buffer avoids the need for a lossless priority group to require a reserved buffer to accept incoming traffic after generating a pause packet.

- A shared buffer is dynamically allocated based on the threshold value for using the available free memory. The dynamic threshold value is a configurable integer (-6 to 3) that determines the shared buffer threshold, and the amount of shared buffer space during congestion events. The shared buffer space is the total buffer pool size minus the reserved buffer space in ingress and egress buffer pools.
- You allocate the amount of buffer memory used by an ingress priority group or an egress queue by assigning it to a buffer pool.
  - If lossless traffic is required on an ingress port, map all traffic on the port to any priority group (PG) besides PG7 which uses a lossless ingress buffer profile. By default, all ingress traffic is assigned to priority group 7 and handled as lossy traffic.
  - If lossless traffic is required on an egress queue, map all traffic to a queue that uses a lossless egress buffer profile.
- By default, all switch ports use a shared buffer, which allocates a guaranteed amount of buffer memory to each ingress port or priority group, and egress port or queue. The amount of shared buffer space allocated by default is platform-specific. To use available pools of buffer memory, you must assign a buffer profile to an ingress priority group or egress queue.
- When the available port buffer reaches the Xoff threshold, the switch generates pause (Xoff) packets for traffic transmission to avoid packet loss, and only accepts incoming traffic in network connections until a peer stops sending traffic. The number of packets that are accepted depends on factors, such as port speed, cable length, pause response delay, and values based on IEEE standards. When the available buffer returns to the Xon (transmit on) threshold, the switch resumes traffic transmission by sending an Xon packet.
- When you enable priority flow control (PFC) on a port, all PFC-enabled queues and priority-groups use buffers from the lossless buffer profile. To configure PFC using QoS maps, see [Priority flow control](#).
- Buffer usage statistics are collected for ingress packets that use ingress pools and for egress packets that use egress pools.
- Different switches have different amounts of default, reserved, and shared memory buffers — see [Configure ingress buffer](#) and [Configure egress buffer](#) .

#### **Packet walk through ingress and egress buffer admission**

The following figure shows how buffer memory is assigned to QoS priority groups that manage ingress traffic admission, and to egress queues that manage egress traffic admission.



**Figure 15. Ingress and Egress Buffer Admission Rules**

#### Ingress buffer: Using Xoff pause and Xon resume frames

For lossless traffic, an ingress buffer uses two settings to control the generation of Xoff and Xon frames.

- An Xoff frame is sent to pause traffic transmission when the available memory in the priority-group buffer is less than the configured pause-threshold xoff size.
- An Xon frame is sent to resume the traffic transmission when the total buffer usage by priority group — including the shared buffer pool, port buffer pool, and PG buffer usage — is less than or equal to the configured Xon threshold, and the available buffer is greater than or equal to the Xoff threshold.

In the ingress buffer, the Xoff pause threshold is set using a part of the PG buffer pool. The Xon resume threshold is set using a part of the shared buffer pool.



**Figure 16. Xoff pause and Xon resume frame generation in ingress buffer**

## Pre-configured NPU defaults

By default, Enterprise SONiC is configured with well-optimized buffers at NPU initialization. By default, all priority groups and queues are configured to use lossy buffers. When a buffer is not available, packet tail drop is performed in all priority groups and queues for congestion avoidance.

The default memory management unit (MMU) configurations for pools, priority groups, and queues are applied before Enterprise SONiC pre-configured buffer settings — see [Pre-configured lossless buffers](#). The MMU defaults are the same as NPU default settings. At NPU initialization, buffers are allocated as follows and can vary for each NPU:

- Ingress traffic: Uses a single pool. The maximum buffer size is calculated by subtracting the reserved buffer sizes. All packets in all traffic classes are mapped to priority group 7 (PG7). PG7 uses a shared static threshold that is set with the maximum buffer size. PG7 is used with no reserved buffers.
- Egress traffic: Uses a single pool. The maximum buffer size is calculated by subtracting the reserved buffer sizes. By default, a guaranteed buffer of 4096 bytes is reserved for all queues on a port. When the guaranteed buffer is exceeded, each egress queue uses the shared buffer pool. Egress queues use a shared buffer with a dynamic threshold value of 3. By default, a reserved buffer of 2048 bytes is reserved for each CPU queue. When this reserved buffer is exceeded, the CPU queues use the shared buffer pool.

**i** **NOTE:** By default, buffer tuning is not supported in Enterprise SONiC. Buffer tuning is supported only in lossless mode after you initialize the pre-configured buffer settings (see [Pre-configured lossless buffers](#)) or enable RoCEv2 (see [Enable RoCEv2 with default configuration](#)).

### Maximum buffer size

**Table 39. Maximum buffer size**

| Enterprise SONiC switch                 | NPU maximum buffer size |
|-----------------------------------------|-------------------------|
| S5200-ON series                         | 32MB                    |
| S5448F-ON                               | 82MB                    |
| Z9264F-ON                               | 42MB                    |
| Z9332F-ON                               | 64MB                    |
| Z9432F-ON                               | 132MB                   |
| Z9664F-ON                               | 113.66MB                |
| N3248TE-ON and E3248P-ON                | 8MB                     |
| N3248PXE-ON, N3248X-ON, and E3248PXE-ON | 32MB                    |

## Pre-configured lossless buffers

Each Enterprise SONiC-supported platform has a configuration file with pre-configured buffer pools, profiles, priority groups, and queues for lossless operation.

**i** **NOTE:** The number of buffer pools depends on the platform/NPU architecture. For example, S5232F-ON, S5248F-ON, and S5296F-ON switches have two egress buffer pools. The Z9332F-ON switch has a single buffer pool.

- A single pool model (ingress\_lossless\_pool and egress\_lossless\_pool) carries both lossy and lossless traffic. A two-pool model (ingress\_lossless\_pool) carries both lossy and lossless traffic.
- The egress\_lossy\_pool is for lossy traffic; the egress\_lossless\_pool is for lossless traffic.
- Although the default buffer pools cannot be deleted or updated, required field modifications are allowed.

To initialize the pre-configured switch-specific QoS buffer settings, enter the `buffer init lossless` command. Enable pre-configured buffers only when lossless operation is needed.

```
sonic(config)# buffer init lossless
```

**i** **NOTE:** To enable RoCEv2 with the default RoCEv2/ISCSI lossless buffer settings and the default WRED/ECN, scheduling, and qos map configurations defined for a switch, use the `roce enable` command — see [Enable RoCEv2 with default configuration](#).

## Usage notes

- You cannot delete or modify pre-configured default pools and profiles. Priority groups 3 and 4, and queues 3 and 4, are reserved for lossless traffic. You can, however, create new buffer profiles and apply them to PFC priority groups and egress queues — see [Configure buffers](#).
- PG7 is dedicated to lossy traffic, and uses the shared static threshold which is set to the maximum buffer size. No packet drop is performed due to ingress admission control. Modifying a buffer profile on PG7 is not supported.
- Queues 3 and 4 are reserved for lossless traffic. The lossless buffer profile, which uses the shared static threshold set to the maximum buffer size, is applied on queues 3 and 4. No packet drop is performed due to egress admission control. Modifying a buffer profile on queues 3 and 4 is not supported.

## View lossless buffer configuration

To view the pre-configured QoS buffer configurations, use the `show buffer pool`, `show buffer profile`, and `show buffer interface` commands.

```
sonic# show buffer pool

egress_lossless_pool:
 size : 31617024
 type : egress
 mode : static

egress_lossy_pool:
 size : 24320512
 type : egress
 mode : dynamic

ingress_lossless_pool:
 size : 32157184
 type : ingress
 shared-headroom-size : 2621440
 mode : dynamic
```

```
sonic# show buffer profile

egress_lossless_profile:
 pool : egress_lossless_pool
 size : 0
 static-threshold : 31617024

egress_lossy_profile:
 pool : egress_lossy_pool
 size : 0
 dynamic-threshold : 3

ingress_lossy_profile:
 pool : ingress_lossless_pool
 size : 0
 static-threshold : 32566016

pg_lossless_25000_40m_profile:
 pool : ingress_lossless_pool
 size : 9216
 dynamic-threshold : 0
 pause-threshold : 67840
 resume-threshold : 9216
 resume-offset-threshold : 9216

pg_lossless_100000_40m_profile:
 pool : ingress_lossless_pool
 size : 9216
 dynamic-threshold : 0
 pause-threshold : 173568
 resume-threshold : 9216
 resume-offset-threshold : 9216
```

```
sonic# show buffer interface Ethernet all priority-group
```

| Interface | priority-group | Profile                       |
|-----------|----------------|-------------------------------|
| Ethernet0 | 3-4            | pg_lossless_25000_40m_profile |
| Ethernet0 | 7              | ingress_lossy_profile         |
| Ethernet1 | 3-4            | pg_lossless_25000_40m_profile |
| Ethernet1 | 7              | ingress_lossy_profile         |
| Ethernet2 | 3-4            | pg_lossless_25000_40m_profile |
| Ethernet2 | 7              | ingress_lossy_profile         |
| Ethernet3 | 3-4            | pg_lossless_25000_40m_profile |
| Ethernet3 | 7              | ingress_lossy_profile         |

|            |     |                                |
|------------|-----|--------------------------------|
| Ethernet4  | 3-4 | pg_lossless_25000_40m_profile  |
| Ethernet4  | 7   | ingress_lossy_profile          |
| Ethernet5  | 3-4 | pg_lossless_25000_40m_profile  |
| Ethernet5  | 7   | ingress_lossy_profile          |
| Ethernet6  | 3-4 | pg_lossless_25000_40m_profile  |
| Ethernet6  | 7   | ingress_lossy_profile          |
| Ethernet7  | 3-4 | pg_lossless_25000_40m_profile  |
| Ethernet7  | 7   | ingress_lossy_profile          |
| Ethernet8  | 3-4 | pg_lossless_25000_40m_profile  |
| Ethernet8  | 7   | ingress_lossy_profile          |
| Ethernet9  | 3-4 | pg_lossless_25000_40m_profile  |
| Ethernet9  | 7   | ingress_lossy_profile          |
| Ethernet10 | 3-4 | pg_lossless_25000_40m_profile  |
| Ethernet10 | 7   | ingress_lossy_profile          |
| ...        |     |                                |
| Ethernet48 | 3-4 | pg_lossless_100000_40m_profile |
| Ethernet48 | 7   | ingress_lossy_profile          |
| Ethernet52 | 3-4 | pg_lossless_100000_40m_profile |
| Ethernet52 | 7   | ingress_lossy_profile          |
| Ethernet56 | 3-4 | pg_lossless_100000_40m_profile |
| Ethernet56 | 7   | ingress_lossy_profile          |
| Ethernet60 | 3-4 | pg_lossless_100000_40m_profile |
| Ethernet60 | 7   | ingress_lossy_profile          |
| Ethernet64 | 3-4 | pg_lossless_100000_40m_profile |
| Ethernet64 | 7   | ingress_lossy_profile          |
| Ethernet68 | 3-4 | pg_lossless_100000_40m_profile |
| Ethernet68 | 7   | ingress_lossy_profile          |
| Ethernet72 | 3-4 | pg_lossless_100000_40m_profile |
| Ethernet72 | 7   | ingress_lossy_profile          |
| Ethernet76 | 3-4 | pg_lossless_100000_40m_profile |
| Ethernet76 | 7   | ingress_lossy_profile          |

```
sonic# show buffer interface Ethernet all queue
```

| Interface  | queue    | Profile                 |
|------------|----------|-------------------------|
| CPU        | 0-47     | egress_lossy_profile    |
| Ethernet0  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet0  | 3-4      | egress_lossless_profile |
| Ethernet1  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet1  | 3-4      | egress_lossless_profile |
| Ethernet2  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet2  | 3-4      | egress_lossless_profile |
| Ethernet3  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet3  | 3-4      | egress_lossless_profile |
| Ethernet4  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet4  | 3-4      | egress_lossless_profile |
| Ethernet5  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet5  | 3-4      | egress_lossless_profile |
| Ethernet6  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet6  | 3-4      | egress_lossless_profile |
| Ethernet7  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet7  | 3-4      | egress_lossless_profile |
| Ethernet8  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet8  | 3-4      | egress_lossless_profile |
| Ethernet9  | 0-2,5-19 | egress_lossy_profile    |
| Ethernet9  | 3-4      | egress_lossless_profile |
| Ethernet10 | 0-2,5-19 | egress_lossy_profile    |
| Ethernet10 | 3-4      | egress_lossless_profile |
| ...        |          |                         |
| Ethernet48 | 0-2,5-19 | egress_lossy_profile    |
| Ethernet48 | 3-4      | egress_lossless_profile |
| Ethernet52 | 0-2,5-19 | egress_lossy_profile    |
| Ethernet52 | 3-4      | egress_lossless_profile |
| Ethernet56 | 0-2,5-19 | egress_lossy_profile    |
| Ethernet56 | 3-4      | egress_lossless_profile |
| Ethernet60 | 0-2,5-19 | egress_lossy_profile    |
| Ethernet60 | 3-4      | egress_lossless_profile |
| Ethernet64 | 0-2,5-19 | egress_lossy_profile    |
| Ethernet64 | 3-4      | egress_lossless_profile |
| Ethernet68 | 0-2,5-19 | egress_lossy_profile    |
| Ethernet68 | 3-4      | egress_lossless_profile |
| Ethernet72 | 0-2,5-19 | egress_lossy_profile    |

|            |          |                         |
|------------|----------|-------------------------|
| Ethernet72 | 3-4      | egress_lossless_profile |
| Ethernet76 | 0-2,5-19 | egress_lossy_profile    |

## Configure buffers

An Enterprise SONiC switch uses multiple buffer pools with reserved and shared memory to minimize packet loss and handle traffic congestion.

**(i) NOTE:** In Enterprise SONiC, reconfiguring or fine tuning buffers is supported only after you initialize the pre-configured switch-specific QoS buffer settings — see [Preconfigured buffers](#). Pre-configured buffers reserve priority groups 3 and 4, and queues 3 and 4 for lossless traffic, and apply lossless settings. Buffer tuning is not supported if you do not enable pre-configured lossless buffers using the `buffer init lossless` command ([Pre-configured lossless buffers](#)) or lossless RoCEv2 operation using the `roce enable` command ([Enable RoCEv2 with default configuration](#)).

**(i) NOTE:** Lossless buffer configuration and tuning is supported only on the S5232F-ON, S5248F-ON, S5296F-ON, and Z9332F-ON platforms.

To reconfigure buffer settings:

1. Populate the platform-specific default configurations for ingress and egress buffer pools, lossy profiles, priority groups, and queues by entering the buffer initialization command — see [Pre-configured buffer pools](#). The `buffer init lossless` command installs the platform-specific default configurations for the lossless buffer profile. Installing the lossless buffer defaults is required for RoCEv2 traffic transmission — see [RDMA over Converged Ethernet](#).

**(i) NOTE:** Each Enterprise SONiC-supported platform has pre-configured switch-specific QoS buffer settings for RoCEv2 operation. The pre-configured buffer settings are initialized when you enable RoCEv2 operation using the `roce enable` command — see [Enable RoCEv2 with default configuration](#). The `buffer init lossless` and `roce enable` commands are mutually exclusive because both commands set the same lossless buffers.

```
sonic(config) # buffer init lossless
```

To uninstall the lossless buffer default setting and return to the lossy buffer default settings, use the `no buffer init` command.

2. If the default shared headroom size (pool xoff) configured in Step 1 is not sufficient or if you want to avoid under-using the reserved shared buffer pool, reconfigure the headroom buffer size. Enter the size (in bytes) of the shared memory pool; the minimum and maximum bytes are platform-dependent. All other buffer pool configurations are not user-configurable. To view the preset default buffer pool and profile configurations, use the `show buffer pool` command.

```
sonic(config) # buffer pool ingress_lossless_pool shared-headroom-size shared-buffer-size-in-bytes
```

For example:

```
sonic(config) # buffer pool ingress_lossless_pool shared-headroom-size 20000
```

3. Configure a buffer profile and associate it with an ingress or egress buffer pool. The buffer profile specifies the guaranteed (reserved) memory for queues or PFC priority groups, the static or dynamic thresholds, and the pause and resume thresholds.

```
sonic(config) # buffer profile buffer-pool-name reserved-buffer-size-in-bytes
{{dynamic-threshold | static-threshold} signed-integer-value} {pause-threshold bytes}
{resume-threshold bytes} {resume-offset-threshold bytes}
```

- `buffer-pool-name` — The valid buffer pool names are: `ingress_lossless_pool`, `egress_lossless_pool`, and `egress_lossy_pool`.
- `reserved-buffer-size-in-bytes` — Enter the reserved buffer size in bytes reserved from the buffer pool (0 to 9216; no default).
- `dynamic-threshold signed-integer-value` — Enter the dynamic size of the buffer threshold by specifying the number of low order bits that can contain data in queued packets. The valid values are -6, -5, -4, -3, -2, -1, 0, 1, 2, and 3; there is no default.
- `static-threshold signed-integer-value` — Enter the fixed, static size in bytes of the maximum threshold used to buffer packets (0 to the maximum NPU buffer size; no default). This is not a reserved buffer.

- `pause-threshold bytes` — (Mandatory for a buffer profile that creates ingress lossless pools; not required for a buffer profile with egress lossless and lossy pools) Enter the number of bytes for the maximum size of the shared headroom buffer used from the ingress pool (1 to the maximum platform-specific ingress pool shared headroom size; no default). An available buffer for a priority group that is less than the specified size triggers the sending of pause frames and is equal to the Xoff or headroom value.
- `resume-threshold bytes` — (Mandatory for a buffer profile that creates ingress lossless pools; not required for a buffer profile egress lossless and lossy pools) Enter the number of bytes for the threshold that is used to resume packet transmission (1 to 18432 bytes; no default).
- `resume-offset-threshold bytes [xon]` — (Mandatory for a buffer profile that creates ingress lossless pools; not required for egress lossless and lossy pools) Enter the number of bytes for the offset value that is used to resume packet transmission (1 to 18432 bytes; no default). Enter `xon` to send a notification to a sending device to indicate that the switch is now ready to accept data.

For example:

```
sonic(config)# buffer profile profile_2 egress-lossy-pool 20000 dynamic-threshold -2
static-threshold 2000

sonic(config)# buffer profile profile_3 egress-lossy-pool 30000 pause pause-threshold
3000 resume-threshold 2000 resume-offset-threshold 200
```

To delete a buffer profile, enter the `no buffer profile name` command.

4. Associate a buffer profile with one or more PFC priority groups on an interface. Separate individual priorities and a priority range with a comma. PG7 is reserved for lossless traffic and does not support profile assignment.

```
sonic(config)# interface Ethslot/port[.breakout]
sonic(conf-if-Eth)# buffer priority-group pg-value-range buffer-profile-name
```

For example:

```
sonic(config)# interface Eth1/1
sonic(conf-if-Ethernet0)# buffer priority-group 3-4 profile_1
```

Disassociating a PFC priority group from a buffer profile is not supported.

5. Associate an egress queue with a buffer profile on an interface. Separate individual queues and a queue range with a comma. Q3 and Q4 are reserved for lossless traffic and do not support profile assignment.

```
sonic(config)# interface Ethslot/port[.breakout]
sonic(conf-if-Eth)# buffer queue queue-value-range buffer-profile-name
```

For example:

```
sonic(config)# interface Ethernet0
sonic(conf-if-Ethernet0)# buffer queue 5-6 profile_2
```

Disassociating an egress queue from a buffer profile is not supported.

6. (Optional) Enable the default lossless buffer profile on an interface. To view the default lossless profile settings, use the `show buffer profile` command.

```
sonic(config)# interface Ethslot/port[.breakout]
sonic(conf-if-Eth)# buffer default-lossless-buffer-profile
```

For example:

```
sonic(config)# interface Ethernet0
sonic(conf-if-Ethernet0)# buffer default-lossless-buffer-profile
```

To disable the default lossless buffer profile, enter the `no buffer default-lossless-buffer-profile` command in Interface configuration mode.

 **NOTE:** To delete all QoS buffer configurations, enter the `no buffer init` command.

```
sonic(config)# no buffer init
```

## Configure ingress buffer

By default, all traffic classes map to the default priority group PG7 for ingress buffers. The buffer reservation is based on the default priority group ID 7. All buffers are part of the default pool and all ports share buffers from the default pool.

**(i) NOTE:** After you enable pre-configured lossless buffers using the `buffer init lossless` command ([Pre-configured lossless buffers](#)) or lossless RoCEv2 operation using the `roce enable` command ([Enable RoCEv2 with default configuration](#)), lossless traffic should be mapped to PG3 and PG4. In lossless mode, PG7 does not have any reserved buffers. PG3 and PG4 support pre-configured buffer profiles for lossless traffic handling and user-defined buffer profiles.

### Default ingress buffer

To view the default ingress buffer configurations on a switch , enter the `show buffer pool` and `show buffer profile` commands. These values may differ for different platforms and speeds.

### PFC priority-group buffer

The default PFC buffer settings for each PFC priority group are platform-dependent. To view the PFC priority group mapping to ingress interfaces, use the `show buffer interface` command.

**(i) NOTE:** The supported speeds vary across Enterprise SONiC platforms. After reserved buffers are used, PFC or PG3 and PG4 start consuming shared buffers from the lossless pool using the dynamic shared buffer threshold.

To reconfigure the default PFC priority group settings, see [Reconfigure RoCEv2 default configurations](#).

## Configure egress buffer

All egress port queues are allocated with reserved buffers. When the reserved buffers are consumed, each queue starts using the shared buffers in the default pool.

When you enable pre-configured lossless buffers using the `buffer init lossless` command ([Pre-configured lossless buffers](#)) or lossless RoCEv2 operation using the `roce enable` command ([Enable RoCEv2 with default configuration](#)), there are no reserved buffers for each egress queue. All queues use shared buffers.

The default dynamic shared buffer threshold for an egress queue is 3. To reconfigure the dynamic shared buffer threshold, configure a buffer profile for the egress lossy pool and apply the profile to the egress queues; for example:

```
sonic(config)# buffer profile q-profile egress-lossy-pool 1000 dynamic-threshold -2
sonic(config)# interface Eth1/2
sonic(config-if-Eth1/2)# buffer queue 0-2 q-profile
```

# RDMA over Converged Ethernet

RDMA over Converged Ethernet (RoCE) provides Remote Direct Memory Access (RDMA) over an Ethernet network, enabling memory transfer between two devices that bypasses the CPU on each device. Encapsulated InfiniBand (IB) transport packets are transmitted over lossless converged Ethernet links between servers and storage devices in a data center network. Enterprise SONiC supports the RoCE protocol versions 1 and 2:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>RoCEv1</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | An Ethernet link-layer protocol that allows only Layer 2 communication between devices in the same VLAN (Ethernet broadcast domain).                                  |
| <b>RoCEv2</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Enhances RoCEv1 by adding Layer 3 routability. RDMA transport packets are encapsulated in an Ethernet/IPv4 or IPv6/UDP header. RoCEv2 is also known as routable RoCE. |
| <b>i   NOTE:</b> Enabling ROCE on an Enterprise SONiC switch enables both ROCEv2 — with the default RoCEv2/iSCSI lossless buffer settings and the default WRED/ECN, scheduling, and qos map configurations defined for the switch — and ROCEv1 operation. However, note that ROCEv1 only supports Layer 2 communication within the same VLAN (Ethernet broadcast domain) with Ethertype 0x8915, using PFC and the dot1p-tc map configuration. ROCEv1 traffic does not use the dscp-tc map configuration. |                                                                                                                                                                       |

## RoCEv2 operation

RoCEv2 ensures zero packet loss with low-latency, high-throughput data transmission for RoCEv2 distributed applications in an Ethernet-based data center network. RoCEv2 accomplishes reliable packet delivery using existing congestion management features, including:

- Priority flow control (PFC) — Prevents packet loss according to class-of-service (CoS) values. When a buffer queue threshold is exceeded, pause frames are sent to request a switch or server sending device to stop sending frames with a specified CoS for a certain time — see [Priority flow control](#). Other priority traffic with different CoS values is not impacted.
  - Enhanced transmission selection (ETS) — Assigns bandwidth to each CoS priority value to prevent some traffic classes from using most of the bandwidth on a link. If CoS traffic does not fully use its allocated bandwidth, the free bandwidth is made available to other CoS traffic classes.
- i | NOTE:** ETS is not supported in 4.1.0 and later releases. By default, RoCEv2 configures bandwidth for lossy and lossless queues.
- Data Center Bridging Exchange (DCBX) protocol — (Optional) Negotiates the PFC, ETS, and CoS values between DCB-capable peer devices on an Ethernet link. Although DCBX is not required for RoCEv2 operation, it simplifies RoCEv2 management. DCBX functions like LLDP with type, length, and value (TLV) fields.
- i | NOTE:** DCBX is not supported in the 4.1.0 and later releases.
- Explicit Congestion Notification (ECN) — As an extension to WRED, ECN avoids dropping packets when a queue threshold is exceeded by sending congestion notifications to sending network devices so that they reduce traffic transmission — see [WRED and ECN](#). Configure WRED ECN on all nodes between the source and destination devices. Whereas PFC is performed per-hop and may take more time to propagate congestion information to a sending device, when ECN detects congestion, it immediately sends notification to all endpoint devices and marks the congested traffic on each network device so that no traffic is dropped.
- i | NOTE:** To handle L3 packets, RoCEv2 requires ECN to be enabled so that the congestion bits in frames of congested traffic are updated. If traffic already experiences congestion, the congestion bits are not updated.

## RoCEv2 prerequisites

- RoCEv2 requires that you configure PFC on the Ethernet links to prevent packet loss due to buffer overflow and ensure lossless data transmission — see [Priority flow control](#).
- RoCEv2 requires RDMA-capable Ethernet NICs (rNICs) on servers and storage devices.
- RoCEv2 is included on top of UDP/IPv4 or UDP/IPv6 in a packet header protocol, and uses UDP destination-port number 4791 by default. RoCEv2 guarantees that packets with the same UDP source port and destination port are delivered in the correct order.

## RoCEv2 configuration notes

- Use RoCEv2 on edge switches in SAN and tenant fabrics to manage converged traffic from downstream servers and storage devices.

- RoCEv2 is supported only on the S5232F-ON, S5248F-ON, S5296F-ON, Z9664F-ON, Z9432F-ON, and Z9332F-ON platforms.
- When RoCEv2 uses PFC and ECN at the same time, ECN functions as the primary congestion management mechanism; PFC is secondary.
- Configure breakout ports before you initialize pre-configured QoS buffer settings using the `buffer init lossless` command (see [Pre-configured lossless buffers](#)) or enable RoCEv2 (see [Enable RoCEv2 with default configuration](#)). Breakout ports that you configure after enabling RoCEv2 are not supported. If you need to configure a breakout port after you enable RoCEv2 or lossless buffers, disable RoCEv2 and the breakout ports, and then re-enable RoCEv2.
- Configure the amount of shared and reserved memory used for ingress priority groups and egress queues as needed — see [Buffer management](#).
- Lossless operation is supported only for unicast traffic. The multicast traffic pause setting is not honored.
- Lossless traffic is tagged — Use lossless traffic (lossless DSCP) with dot1p priority 3 and 4. If you are using DSCP-based PFC, map lossless traffic with DSCP values mapped to traffic class 3 to VLAN dot1p priority 3, and map lossless traffic with DSCP values mapped to traffic class 4 to dot1p priority 4.
  - The priority group (PG) function sets the buffer required to generate PFC pause frames. When a PG detects congestion, PFC transmit (tx) pause frames are generated with a PFC priority value. In many deployments, a single traffic class (TC) or a packet's dot1p is mapped to one priority group; for example, TC3 to PG3 and TC4 to PG4. You can also map multiple traffic classes to a single priority group; for example, TC3 and TC4 to PG3. In this case, when PG3 detects congestion, a pause frame is generated with both PFC priorities 3 and 4. This behavior, however, is not recommended because if one flow is congested, the other flow is also affected and is treated as being congested.
  - To generate PFC pause frames, you must enable PFC priority values on an interface using the `priority-flow-control priority` command - see [Priority flow control](#). Enterprise SONiC only supports PFC priorities 3 and 4.
  - When you enable PFC for a specified PFC priority value, Enterprise SONiC requires that the PFC priority value matches the configured traffic class (TC) values in each `qos map`. In PFC configuration, ingress traffic is classified as follows:
    - Assign dot1p and DSCP values in ingress packets to a traffic class — `qos map dot1p-tc` and `qos map dscp-tc`.commands.
    - Assign ingress traffic classes to a priority group — `qos map tc-pg`.command.
    - Assign ingress traffic classes to a priority queue — `qos map tc-queue`.command.

For VLAN-tagged packets, the packet's dot1p value takes precedence over the TC-to-PG mapping. To provide uniform behavior for tagged and untagged ROCE flows, Dell Technologies recommends that you configure the same TC and Dot1p values when enabling a PFC priority value. For example, for ROCE operation, configure dot1p 3 and tc 3 or dot1p 5 and tc 5 in qos maps before you enable `priority-flow-control priority 3`. By default, Enterprise SONiC supports lossless buffer settings on PG3 and PG4. You can modify the buffer configurations only on PG3 and PG4, and only for the PFC priority (3 or 4) that is enabled on an interface. On an Enterprise SONiC switch, you must map ROCE traffic to dot1p 3.

- On servers, Dell Technologies recommends that you set the DSCP value 48 for congestion notification packets (CNP), if possible. This setting allows a RoCEv2-enabled switch to prioritize the downstream server traffic it receives by applying strict priority queuing.
- The default RoCEv2 configuration is only applied on queue 3 for ECN-treated traffic. If a second RoCEv2-supported queue is needed, Dell Technologies recommends that you configure ECN on queue 4.
- The default RoCEv2 configuration reserves two lossless queues (queues 3 and 4; traffic classes 3 and 4) on each port. If only one lossless queue is required, Dell Technologies recommends that you do not classify traffic to the required traffic classes 3 and 4.
- Dell Technologies recommends you handle multicast traffic as follows: Either do not manage multicast traffic on queues 3 and 4 or use interface storm control to drop multicast packets if multicast is not required in a data center.
- If a port is a member of a port channel, apply the QoS maps that are supported on a port channel (`dscp-tc`, `dot1p-tc`, `tc-dscp`, `tc-dot1p` — see [QoS maps](#)) on the port channel, not on member ports. The default QoS maps for `dscp-tc` and `dot1p-tc` are applied on the port channel.
- Lossless operation is not supported on multicast and destination lookup failure (DLF) packets — unknown unicast and broadcast; this traffic is treated as lossy traffic. Dell Technologies recommends that you do not configure multicast and DLF traffic with the same lossless DSCP or dot1p value. To control traffic storms on an interface, use storm control commands:

```
sonic(config-if-Ethernet0) # storm-control {broadcast | unknown-multicast | unknown-unicast}
```

- In the default RoCEv2 configurations, PFC watchdog is enabled on all ports — see [Priority flow control](#).
- In the default RoCEv2 configurations, asymmetric PFC is not enabled.
- When RoCEv2 with the default configuration (`roce enable` command) or enable pre-configured lossless buffers (`buffer init lossless` command) is enabled so that a switch operates in lossless mode, all dot1p priorities in ingress traffic are honored regardless of the PFC priority configuration.

- On Z9432F-ON and Z9664F-ON switches, the `show buffer_pool {watermark | persistent-watermark}` commands are not supported.

## Topics:

- [Enable RoCEv2 with default configuration](#)
- [View RoCEv2 default configurations](#)
- [Reconfigure RoCEv2 default configurations](#)
- [RoCE traffic hashing](#)

## Enable RoCEv2 with default configuration

To enable RoCEv2 with the default RoCEv2/ISCSI lossless buffer settings and the default WRED/ECN, scheduling, and `qos map` configurations defined for a switch, use the `roce enable` command. You are prompted to save the default configuration and restart the switch.

- Enable lossless RoCEv2 transmission globally on the switch using the default QoS and lossless buffer configurations. Enter `y` to save the default configurations and restart the switch.

```
sonic(config) # roce enable
This command will also restart the node after saving all configurations. [Proceed
y/N]: y
```

To remove the default QoS and all buffer initializations, use the `no roce enable` command.

- i** **NOTE:** After you enable RoCEv2 pre-configured default settings, if pre-existing configurations are already present on the switch that conflict with the default configurations (for example, `dscp-tc` and `dot1p-tc` maps, scheduler settings, WRED on queue 3 or 4), an error message is displayed. Dell Technologies recommends that you delete the previously configured settings and enable the RoCEv2 defaults using the `roce enable` command, or enter the `roce enable force-defaults` command, which removes conflicting configurations and applies RoCEv2 defaults. This command clears any previously applied QoS buffers and forces an initialization of the RoCEv2 default buffer configuration.

```
sonic(config) # roce enable force-defaults
```

After the switch reboots, use the `show qos` and `show buffer` commands to display the default RoCEv2 configurations — see [View RoCEv2 default configurations](#).

- i** **NOTE:** Instead of using the default RoCEv2 configurations, you can also use RoCEv2 by manually configuring traffic classes, priority flow control, WRED/ECN policy, scheduler, PFC watchdog, and lossless buffers for RoCEv2 — see [Priority flow control, WRED and ECN, Scheduler policy](#), and [Buffer management](#). Then enable the RoCEv2 configurations on the switch.

## View RoCEv2 default configurations

After you enable RoCEv2 with the default RoCEv2/ISCSI lossless buffer settings and the default WRED/ECN, scheduling , and `qos map` configurations that are defined for a switch, use `show qos` and `show buffer` commands to view the default configurations.

### View default RoCEv2 traffic classes and PFC mapping

Ingress DSCP and dot1p CoS values are mapped to the RoCE traffic class. If both the DSCP and dot1p values are present in a packet header, the DSCP value is used to classify traffic. The dot1p value in the packet header is used to assign the priority-group; tc-pg mapping is ignored — see [Traffic class to dot1p map](#). For tagged frames, Dell Technologies recommends that you always use dot1p 3 for lossless traffic mapped to traffic class 3, and dot1p 4 for lossless traffic mapped to traffic class 4.

- Widely used ISCSI DSCP 4 is mapped to traffic-class 4.
- Well-known or widely used RDMA DSCP values 24 and 26 are mapped to traffic-class 3.
- In CNP packets, DSCP 48 is mapped to traffic-class 6.

- All other DSCP values are mapped to traffic-class 0.

```
sonic# show qos map dscp-tc
DSCP-TC-MAP: ROCE

 DSCP TC

 0 0
 1 0
 2 0
 3 0
 4 4
 5 0
...
 23 0
 24 3
 25 0
 26 3
 27 0
 28 0
...
 46 0
 47 0
 48 6
...
 62 0
 63 0
```

```
sonic# show qos map dot1p-tc
DOT1P-TC-MAP: ROCE

 DOT1P TC

 0 0
 1 0
 2 0
 3 3
 4 4
 5 0
 6 0
 7 0
```

Ingress traffic classes are assigned to RoCEv2 PFC priority groups.

```
sonic# show qos map tc-pg
Traffic-Class-Priority-Group-MAP: ROCE

 TC PG

 0 7
 1 7
 2 7
 3 3
 4 4
 5 7
 6 7
 7 7
```

PFC priority traffic is assigned to RoCEv2 PFC priority queues.

```
sonic# show qos map pfc-priority-queue
PFC-Priority-Queue-MAP: ROCE

 PFC Priority Queue

 0 0
 1 1
 2 2
 3 3
 4 4
 5 5
```

### View default RoCEv2 WRED/ECN policy

The default scheduler policy for WRED/ECN configures the PFC priority queues for RoCEv2 traffic. The default scheduler gives equal weight to lossy and lossless traffic, and assigns strict priority to CNP packets.

```
sonic# show qos scheduler-policy
Scheduler Policy: ROCE
Queue: 0
 type: dwrr
 weight: 50
Queue: 3
 type: dwrr
 weight: 50
Queue: 4
 type: dwrr
 weight: 50
Queue: 6
 type: strict
```

The default WRED policy configures minimum and maximum green threshold values, maximum drop rate, and the ECN traffic filter.

```
sonic# show qos wred-policy

Policy : ROCE

ecn : ecn_green
green-min-threshold : 1048 KBytes
green-max-threshold : 2097 KBytes
green-drop-probability : 5
```

### View default RoCEv2 buffer configurations

By default, a total amount of available memory (in bytes) is assigned to ingress and egress buffer pools. The egress lossless pool is reserved with a fixed amount of memory. The egress lossy and ingress lossless pools are shared buffers with dynamically allocated memory based on a threshold value for using free memory. The ingress lossless pool displays the size of its shared memory.

**i** **NOTE:** The number of buffer pools depends on the platform/NPU architecture. For example, Z9332F-ON, S5232F-ON, S5248F-ON, and S5296F-ON switches have two egress buffer pools. The Z9332F-ON, Z9432F-ON, and Z9664F-ON switches have a single buffer pool.

- A single pool model (ingress\_lossless\_pool and egress\_lossless\_pool) carries both lossy and lossless traffic. A two-pool model (ingress\_lossless\_pool) carries both lossy and lossless traffic.
- The egress\_lossy\_pool is for lossy traffic; the egress\_lossless\_pool is for lossless traffic.
- Although the default buffer pools cannot be deleted or updated, required field modifications are allowed.

```
sonic# show buffer pool

egress_lossless_pool:
 size : 31617024
 type : egress
 mode : static
egress_lossy_pool:
 size : 24320512
 type : egress
 mode : dynamic
ingress_lossless_pool:
 size : 32157184
 type : ingress
 shared-headroom-size : 2621440
 mode : dynamic
```

Various buffer profiles are created and associated with an ingress or egress buffer pool. A buffer profile specifies the guaranteed (reserved) memory for queues or PFC priority groups, static or dynamic thresholds, and optional pause and resume thresholds.

```
sonic# show buffer profile
egress_lossless_profile:
 pool : egress_lossless_pool
 size : 0
 static-threshold : 67117468
egress_lossy_cpu_profile:
 pool : egress_lossless_pool
 size : 9144
 dynamic-threshold : -5
egress_lossy_profile:
 pool : egress_lossless_pool
 size : 0
 dynamic-threshold : 2
ingress_lossy_profile:
 pool : ingress_lossless_pool
 size : 0
 static-threshold : 66394076
pg_lossless_10000_5m_profile:
 pool : ingress_lossless_pool
 size : 9144
 dynamic-threshold : -2
 pause-threshold : 58674
 resume-threshold : 9144
 resume-offset-threshold : 9144
pg_lossless_10000_40m_profile:
 pool : ingress_lossless_pool
 size : 9144
 dynamic-threshold : -2
 pause-threshold : 59944
 resume-threshold : 9144
 resume-offset-threshold : 9144
pg_lossless_10000_300m_profile:
 pool : ingress_lossless_pool
 size : 9144
 dynamic-threshold : -2
 pause-threshold : 69596
 resume-threshold : 9144
 resume-offset-threshold : 9144
```

### View default RoCEv2 configurations per-interface

By default, switch interfaces are assigned to PFC priority groups with ingress buffer profiles.

```
sonic# show buffer interface Ethernet all priority-group
Interface priority-group Profile
Ethernet0 3-4 pg_lossless_25000_40m_profile
Ethernet0 7 ingress_lossy_profile
Ethernet1 3-4 pg_lossless_25000_40m_profile
Ethernet1 7 ingress_lossy_profile
Ethernet2 3-4 pg_lossless_25000_40m_profile
Ethernet2 7 ingress_lossy_profile
Ethernet3 3-4 pg_lossless_25000_40m_profile
Ethernet3 7 ingress_lossy_profile
Ethernet4 3-4 pg_lossless_25000_40m_profile
Ethernet4 7 ingress_lossy_profile
...
Ethernet60 3-4 pg_lossless_100000_40m_profile
Ethernet60 7 ingress_lossy_profile
Ethernet64 3-4 pg_lossless_100000_40m_profile
Ethernet64 7 ingress_lossy_profile
Ethernet68 3-4 pg_lossless_100000_40m_profile
Ethernet68 7 ingress_lossy_profile
Ethernet72 3-4 pg_lossless_100000_40m_profile
Ethernet72 7 ingress_lossy_profile
Ethernet76 3-4 pg_lossless_100000_40m_profile
Ethernet76 7 ingress_lossy_profile
```

By default, switch interfaces are assigned to egress queues with egress buffer profiles.

```
sonic# show buffer interface Ethernet all queue

 Interface queue Profile
CPU 0-47 egress_lossy_profile
Ethernet0 0-2,5-19 egress_lossy_profile
Ethernet0 3-4 egress_lossless_profile
Ethernet1 0-2,5-19 egress_lossy_profile
Ethernet1 3-4 egress_lossless_profile
Ethernet2 0-2,5-19 egress_lossy_profile
Ethernet2 3-4 egress_lossless_profile
Ethernet3 0-2,5-19 egress_lossy_profile
Ethernet3 3-4 egress_lossless_profile
Ethernet4 0-2,5-19 egress_lossy_profile
Ethernet4 3-4 egress_lossless_profile
...
Ethernet60 0-2,5-19 egress_lossy_profile
Ethernet60 3-4 egress_lossless_profile
Ethernet64 0-2,5-19 egress_lossy_profile
Ethernet64 3-4 egress_lossless_profile
Ethernet68 0-2,5-19 egress_lossy_profile
Ethernet68 3-4 egress_lossless_profile
Ethernet72 0-2,5-19 egress_lossy_profile
Ethernet72 3-4 egress_lossless_profile
Ethernet76 0-2,5-19 egress_lossy_profile
```

To view all default QoS settings that are assigned to a specified interface for RoCEv2 traffic, use the `show qos interface` command.

```
sonic# show qos interface Ethernet 0

scheduler policy: ROCE
dscp-tc-map: ROCE
dot1p-tc-map: ROCE
tc-queue-map: ROCE
tc-pg-map: ROCE
pfc-priority-queue-map: ROCE
pfc-asymmetric: off
pfc-priority : 3,4
PFC Watchdog
 Status : on
 Action : drop
 Detection Time : 200ms
 Restoration Time : 400ms
```

```
sonic# show qos interface Ethernet all
| | | |
| Scheduler| Interface Maps | Priority-Flow-Control
| | | |----- WATCHDOG -----
Interface | Policy| dscp-fq dot1p-fq fg-queue fg-pg fg-dscp fg-dot1p pfc-p2q|mode priority action detect restore
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Ethernet0 | ROCE| ROCE ROCE ROCE ROCE ROCE| off 3,4 DROP 200 400
Ethernet1 | ROCE| ROCE ROCE ROCE ROCE ROCE| off 3,4 DROP 200 400
Ethernet2 | ROCE| ROCE ROCE ROCE ROCE ROCE| off 3,4 DROP 200 400
Ethernet3 | ROCE| ROCE ROCE ROCE ROCE ROCE| off 3,4 DROP 200 400
Ethernet4 | ROCE| ROCE ROCE ROCE ROCE ROCE| off 3,4 DROP 200 400
Ethernet5 | ROCE| ROCE ROCE ROCE ROCE ROCE| off 3,4 DROP 200 400
```

### View default RoCEv2 settings in running configuration

To view all of the default QoS settings configured on the switch by enabling RoCEv2, use the `show running configuration` command.

```
sonic# show running-configuration
!
roce enable
!
qos map dscp-tc ROCE
dscp 0-3,5-23,25,27-47,49-63 traffic-class 0
dscp 24,26 traffic-class 3
dscp 4 traffic-class 4
dscp 48 traffic-class 6
!
```

```

qos map dot1p-tc ROCE
 dot1p 0-2,5-7 traffic-class 0
 dot1p 3 traffic-class 3
 dot1p 4 traffic-class 4
!
qos map tc-queue ROCE
 traffic-class 0 queue 0
 traffic-class 1 queue 1
 traffic-class 2 queue 2
 traffic-class 3 queue 3
 traffic-class 4 queue 4
 traffic-class 5 queue 5
 traffic-class 6 queue 6
 traffic-class 7 queue 7
!
qos map tc-pg ROCE
 traffic-class 3 priority-group 3
 traffic-class 4 priority-group 4
 traffic-class 0-2,5-7 priority-group 7
!
qos map pfc-priority-queue ROCE
 pfc-priority 0 queue 0
 pfc-priority 1 queue 1
 pfc-priority 2 queue 2
 pfc-priority 3 queue 3
 pfc-priority 4 queue 4
 pfc-priority 5 queue 5
 pfc-priority 6 queue 6
 pfc-priority 7 queue 7
!
!
qos wred-policy ROCE
 green minimum-threshold 1048 maximum-threshold 2097 drop-probability 5
 ecn green
!
qos scheduler-policy ROCE
!
queue 0
 type dwrr
 weight 50
!
queue 3
 type dwrr
 weight 50
!
queue 4
 type dwrr
 weight 50
!
queue 6
 type strict
!
interface Ethernet0
 mtu 9100
 speed 25000
 unreliable-los auto
 no shutdown
 queue 0 wred-policy test-s
 queue 3 wred-policy ROCE
 scheduler-policy ROCE
 qos-map dscp-tc ROCE
 qos-map dot1p-tc ROCE
 qos-map tc-queue ROCE
 qos-map tc-pg ROCE
 qos-map pfc-priority-queue ROCE
 priority-flow-control priority 3
 priority-flow-control priority 4
 priority-flow-control watchdog action drop
 priority-flow-control watchdog on detect-time 200
 priority-flow-control watchdog restore-time 400
!

```

# Reconfigure RoCEv2 default configurations

After you enable RoCEv2 with the default configurations (see [Enable RoCEv2 with default configuration](#)), you can reconfigure many of the default settings. To display the default RoCEv2 configurations, see [View RoCEv2 default configurations](#).

- (i) NOTE:** If you modify the default RoCEv2 settings — QoS maps, scheduler, WRED — it is your responsibility to maintain the new configuration. It is not possible to revert to the RoCEv2 defaults values without clearing all RoCEv2 QoS configurations and deleting all lossless buffer initializations by using the `no roce enable` command, and then re-enabling the RoCEv2 defaults with the `roce enable` command.

## Reconfigure RoCEv2 traffic classes and QoS PFC mapping

- Re-assign ingress DSCP and/or an dot1p CoS value to a RoCEv2 traffic class. If both the DSCP and dot1p values are present in a packet header, the DSCP value takes precedence over the dot1p value to classify the traffic.

```
sonic(config)# qos map dscp-tc roce-dscp-to-traffic-class-name
sonic(conf-dscp-tc-map-rocev2_dscp-to-tc)# dscp dscp-value traffic-class traffic-class-value
```

```
sonic(config)# qos map dot1p-tc roce-dot1p-to-traffic-class-name
sonic(conf-dot1p-tc-map-rocev2_dot1p-to-tc)# dot1p dot1p-value traffic-class traffic-class-value
```

Examples:

```
sonic(config)# qos map dscp-tc roce_dscp-to-tc
sonic(conf-dscp-tc-map-rocev2_dscp-to-tc)# dscp 6 traffic-class 4
sonic(conf-dscp-tc-map-rocev2_dscp-to-tc)# exit
```

```
sonic(config)# qos map dot1p-tc roce_dot1p-to-tc
sonic(conf-dot1p-tc-map-rocev2_dot1p-to-tc)# dot1p 4 traffic-class 4
sonic(conf-dot1p-tc-map-rocev2_dot1p-to-tc)# exit
```

- Re-assign ingress traffic classes to a RoCEv2 egress or transmit queue. The traffic class of incoming traffic is mapped to the specified transmit queue on an egress interface.

```
sonic(config)# qos map tc-queue roce-tc-queue-name
sonic(conf-qos-map)# traffic-class traffic-class-value queue queue-value
```

For example:

```
sonic(config)# qos map tc-queue roce_tc-q
sonic(conf-tc-queue-map-roce_tc-q)# traffic-class 4 queue 4
sonic(conf-tc-queue-map-roce_tc-q)# exit
```

- Re-assign ingress classes to a RoCEv2 PFC priority group. Only PFC pause frames with priority group 3 and 4 are supported.

```
sonic(config)# qos map tc-pg roce-tc-pg-name
sonic(conf-tc-pg-map-rocev2_tc-pg)# traffic-class traffic-class-value priority-group priority-group-value
```

For example:

```
sonic(config)# qos map tc-pg roce_tc-pg
sonic(conf-tc-pg-map-rocev2_tc-pg)# traffic-class 3 priority-group 3
sonic(conf-tc-pg-map-rocev2_tc-pg)# traffic-class 4 priority-group 4
sonic(conf-tc-pg-map-rocev2_tc-pg)# exit
```

- Re-assign PFC priority traffic to a RoCEv2 PFC priority queue.

```
sonic(config)# qos map pfc-priority-queue rocev2-pfc-priority-queue
sonic(conf-pfc-priority-queue-map-rocev2-pfc-priority-queue)# pfc-priority priority-group-value queue queue-value
```

For example:

```
sonic(config)# qos map pfc-priority-q rocev2_pfc-priority-q
sonic(conf-pfc-priority-queue-map-rocev2_pfc-priority-q)# pfc-priority 3 queue 3
sonic(conf-pfc-priority-queue-map-rocev2_pfc-priority-q)# pfc-priority 4 queue 4
sonic(conf-pfc-priority-queue-map-rocev2_pfc-priority-q)# exit
```

**(i) NOTE:** In addition to modifying default RoCEv2 traffic classes and QoS mapping, you can also delete a default traffic class or QoS mapping, create a non-default traffic class or map, and apply it to an interface. For more detailed information, see [Quality maps](#).

### Reconfigure WRED/ECN policy

- To prevent delay and packet dropping in RoCEv2 transmission, reconfigure the default WRED-ECN policy to early detect and manage queue congestion — see [WRED and ECN](#). Reset the bandwidth in Kilobytes (KB) per second allocated to the minimum and maximum threshold values, and the maximum drop rate for the green traffic class.

```
sonic(config)# qos wred-policy wred-policy-name
sonic(conf-wred-rocev2_wred)# green minimum-threshold minimum-threshold-value maximum-threshold
maximum-threshold-value drop-probability drop-probability-value
```

For example:

```
sonic(config)# qos wred-policy rocev2_wred
sonic(conf-wred-rocev2_wred)# green minimum-threshold 1048 maximum-threshold 2097
drop-probability 5
sonic(conf-wred-rocev2_wred)# ecn all
sonic(conf-wred-rocev2_wred)# exit
```

- Reconfigure the scheduler policy for a queue and WRED policy.

```
sonic(conf-sched-policy)# qos scheduler-policy scheduler-policy-name
sonic(conf-sched-policy)# meter-type packets-or-bytes
sonic(conf-sched-policy)# queue qid {0-7}
sonic(conf-sched-policy-queue-q1)# type {strict | dwrr | wrr}
sonic(conf-sched-policy-queue-q1)# cir cir-value
sonic(conf-sched-policy-queue-q1)# cbs cbs-value
sonic(conf-sched-policy-queue-q1)# pir pir-value
sonic(conf-sched-policy-queue-q1)# pbs pbs-value
```

Re-apply the WRED policy to an interface queue.

```
sonic(conf-sched-policy-queue-q1)# end
sonic# config terminal
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# queue queue-number wred-policy wred-name
```

**(i) NOTE:** In addition to modifying default RoCEv2 WRED/ECN and scheduler policies, you can also delete a default policy and create non-default policies and apply them to an interface. For more detailed information, see [WRED and ECN](#) and [Scheduler policy](#).

### Reconfigure PFC watchdog on an interface

```
sonic(config-if-Ethernet0)#priority-flow-control watchdog action forward
sonic(config-if-Ethernet0)# priority-flow-control watchdog on detect-time 400
sonic(config-if-Ethernet0)# priority-flow-control watchdog restore-time 800
```

### Reconfigure memory buffers

In the default RoCEv2 buffer configurations, you can reconfigure only the shared memory buffer used in the ingress lossless pool. All other default buffer pool configurations are not user-configurable.

```
sonic(config)# buffer pool ingress_lossless_pool shared-headroom-size shared-headroom-size-in-bytes
```

For example:

```
sonic(config)# buffer pool ingress_lossless_pool shared-headroom-size 20000
```

Although you cannot modify or delete default RoCEv2 buffer profiles, you can create a new buffer profile, disassociate the corresponding default buffer profile, then associate the new profile with one or more PFC priority groups and egress queues on an interface. For more detailed information, see [Configure buffers](#). For example:

```
sonic(config)# buffer profile test-pg ingress_lossless_pool 9216 dynamic-threshold 2
pause-threshold 100000 resume-threshold 1024 resume-offset-threshold 1024

sonic(config)# buffer profile test-q egress_lossy_pool 0 dynamic-threshold 0

! Apply buffer profiles to Priority Group/Queue
sonic(config-if-Ethernet0)# no buffer default-lossless-buffer-profile
sonic(config-if-Ethernet0)# buffer priority-group 3-4 test-pg
sonic(config-if-Ethernet0)# buffer queue 0 test-q

sonic(config-if-Ethernet0)# show configuration
!
interface Ethernet0
 mtu 9100
 speed 25000
 no unreliable-los
 shutdown
 buffer priority-group 3 test-pg
 buffer priority-group 4 test-pg
 buffer queue 0 test-q
 no buffer default-lossless-buffer-profile
 queue 3 wred-policy ROCE
 scheduler-policy ROCE
 qos-map dscp-tc ROCE
 qos-map dot1p-tc ROCE
 qos-map tc-queue ROCE
 qos-map tc-pg ROCE
 qos-map pfc-priority-queue ROCE
 priority-flow-control priority 3
 priority-flow-control priority 4
```

In the default RoCEv2 buffer configurations, you can reconfigure only the headroom memory buffer used in the ingress lossless pool. All other default buffer pool configurations are not user-configurable.

```
sonic(config)# buffer pool ingress_lossless_pool shared-headroom-size shared-headroom-
size-in-bytes
```

For example:

```
sonic(config)# buffer pool ingress_lossless_pool shared-headroom-size 20000
```

 **NOTE:** Buffer tuning requires expert-level knowledge. To tune buffers, contact Dell Tech Support — see [Support resources](#).

## RoCE traffic hashing

In 4.2.0 and later releases, an enhanced hashing algorithm provides better load-balancing for ROCE traffic distribution across multiple links. ROCE hashing optimizes the use of the available network bandwidth so that data packets are evenly distributed across available paths, preventing congestion on any single link.

ROCE traffic — RDMA packets encapsulated with a UDP header — are often divided into different flows based on the queue pair (QP) number between the source and destination addresses. In these flows, the regular hashing field attributes, such as the source and destination MAC and IP addresses and source and destination ports, are the same value and do not provide adequate load balancing. ROCE traffic hashing includes the QP number with the native packet field values in the hashing computation.

 **NOTE:** Non-ROCE traffic continues to use regular, native hashing parameters on the Z9664F-ON and other Tomahawk4 platforms.

### Benefits of ROCE traffic hashing

Artificial Intelligence (AI) models process vast amounts of data and require effective algorithms to learn patterns, make predictions, and generate results. Large high-bandwidth data transfers, such as ROCE traffic, are necessary to ensure the success of AI models. ROCE hashing is an effective traffic load distribution method between ECMP and port-channel members

by using the native packet fields, such as source MAC and IP address, destination MAC and IP address, TCP/UDP port numbers, and the QP number.

# Switch protection

|                                                 |                                                                                                                                                                            |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Bidirectional forwarding detection (BFD)</b> | Provides rapid failure detection in links with adjacent routers; see <a href="#">Bidirectional Forwarding Detection</a> .                                                  |
| <b>Link state tracking</b>                      | Associates the loss of upstream connectivity with downstream interfaces that are connected to servers; see <a href="#">Link state tracking</a> .                           |
| <b>Unidirectional link detection (UDLD)</b>     | Detects unidirectional link failures as required in spanning-tree topologies to avoid loops; see <a href="#">Unidirectional link detection</a> .                           |
| <b>Link-error port disabling</b>                | Prevents excessive interface flapping events from adversely affecting routing protocols and routing tables in the network; see <a href="#">Link-error port disabling</a> . |

## Topics:

- [Bidirectional Forwarding Detection](#)
- [Link state tracking](#)
- [Unidirectional link detection](#)
- [Link-error port disabling](#)

## Bidirectional Forwarding Detection

Bidirectional forwarding detection (BFD) rapidly detects communication failures between two adjacent routers. BFD replaces link-state detection mechanisms in existing routing protocols. It also provides a failure detection solution for links with no routing protocols.

BFD provides forwarding-path failure detection in milliseconds instead of seconds. Because BFD is independent of routing protocols, it provides consistent network failure detection. BFD eliminates multiple protocol-dependent timers and methods. Networks converge is faster because BFD triggers link-state changes in the routing protocol sooner and more consistently.

BFD is a simple hello mechanism. Two neighboring routers running BFD establish a session using a three-way handshake. After the session is established, the routers exchange periodic control packets at subsecond intervals. If a router does not receive a hello packet within the specified time, routing protocols are notified that the forwarding path is down.

In addition, BFD sends a control packet when there is a state change or change in a session parameter. These control packets are sent without regard to transmit and receive intervals in a routing protocol.

BFD is an independent and generic protocol, which all media, topologies, and routing protocols can support using any encapsulation. Enterprise SONiC implements BFD at Layer 3 (L3) and with user datagram protocol (UDP) encapsulation. BFD is supported on static and dynamic routing protocols, such as BGP, OSPFv2, and PIM only. The system displays BFD state change notifications.

 **NOTE:** BFD is not supported for static routes, VRRP, and OSPFv3.

### BFD passive mode

When you enable BFD passive mode, the local system does not initiate a BFD session. When the local system receives a BFD control packet from a peer device, the local device responds to the request. By default, BFD passive mode is disabled.

### TTL for multi-hop peer

You can configure a minimum TTL value for a multi-hop BFD peer. If the TTL of the received BFD packet is less than the configured TTL, the system discards the packet. The default TTL value for multi-hop peer is 254.

## BFD session states

To establish a BFD session between two routers, enable BFD on both sides of the link. BFD routers can operate in active role. The active router starts the BFD session. Both routers can be active in the same session.

A BFD session can occur in Asynchronous mode as Enterprise SONiC BFD supports only Asynchronous mode. In Asynchronous mode, both systems send periodic control messages at a specified interval to indicate that their session status is Up.

A BFD session can have four states: Administratively Down, Down, Init, and Up. The default BFD session state is Down.

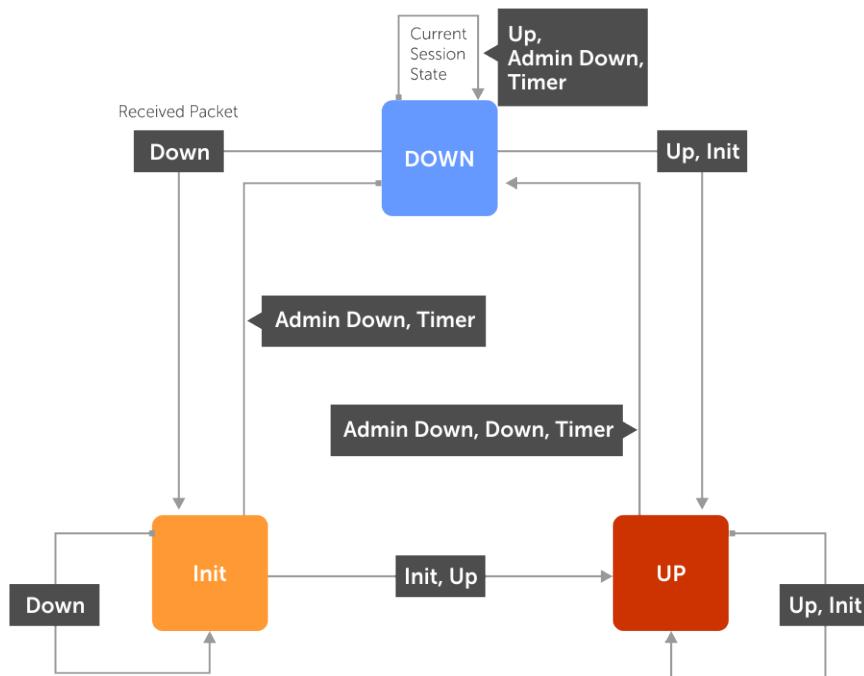
- Administratively Down — The local BFD router does not participate in the session.
- Down — The remote BFD router is not sending control packets or does not send them within the detection time for the session.
- Init — The local BFD router is communicating to the remote router in the session.
- Up — Both BFD routers are sending control packets.

A BFD session's state changes to Down if:

- A control packet is not received within the detection time.
- Demand mode is active and a control packet is not received in response to a poll packet.

### BFD session state changes example

The session state on a router changes according to the status notification it receives from the peer router. For example, if the current session state is Down and the router receives a Down status notification from the remote router, the session state on the local router changes to Init.

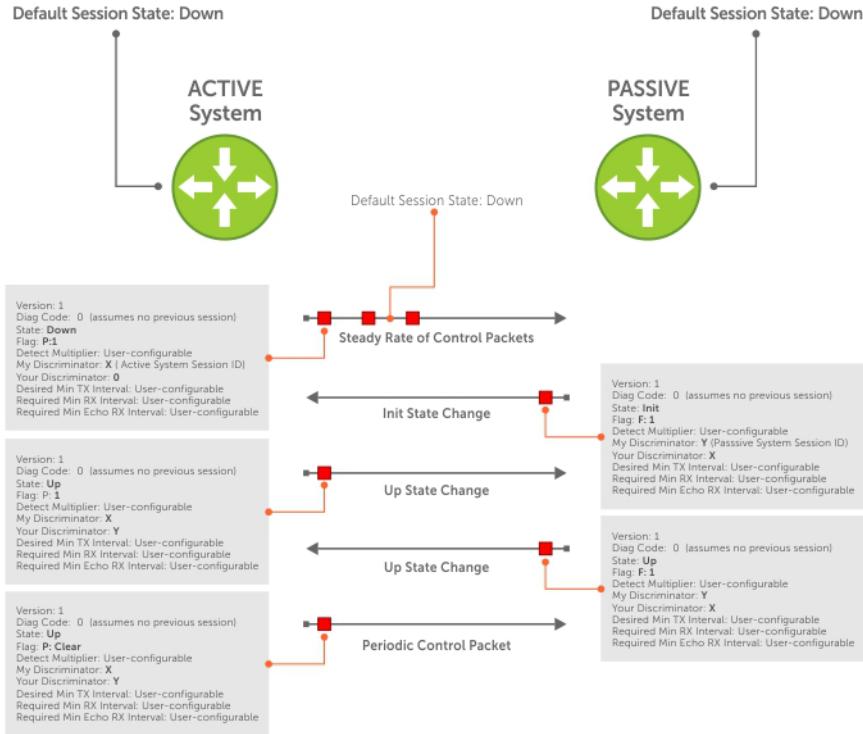


## Three-way handshake

A BFD session requires a three-way handshake between neighboring routers. In this example, the handshake assumes:

- One router is active, and the other router is passive.
  - This is the first session established on this link.
  - The default session state on both ports is Down.
1. The active system sends a steady stream of control packets to indicate that its session state is Down until the passive system responds. These packets are sent at the desired transmit interval of the Active system. The Your Discriminator field is set to one second.
  2. When the passive system receives a control packet, it changes its session state to Init and sends a response to indicate its state change. The response includes its session ID in the My Discriminator field and the session ID of the remote system in the Your Discriminator field.

3. The active system receives the response from the passive system and changes its session state to Up. It then sends a control packet to indicate this state change. Discriminator values exchange, and transmit intervals negotiate.
4. The passive system receives the control packet and changes its state to Up. Both systems agree that a session is established. Because both members must send a control packet, which requires a response only when the session is Up, whenever there is a state change or change in a session parameter, the passive system sends a final response indicating the state change. After this, periodic control packets exchange.



## BFD configuration

Before you configure BFD for a routing protocol, first enable BFD on both routers in the link. BFD is disabled by default.

- Supports 128 BFD sessions with 300 ms intervals and a multiplier of three
- Does not support Demand mode or authentication.
- Does support BFD on multihop sessions.
- Supports protocol liveness only for routing protocols.
- BFD supports BGP, OSPF, and PIM; default and user VRFs are also supported.

## Configure BFD

Before you configure BFD for static routing or a routing protocol, configure BFD on each router, including the BFD session settings. BFD is disabled by default.

- Enable BFD globally.

```
sonic(config) # bfd
```

### View BFD configuration

```
sonic# show running-configuration bfd
!
bfd
peer 192.168.2.1 interface Eth1/1
detect-multiplier 5
```

```

echo-interval 200
echo-mode
receive-interval 200
transmit-interval 200
!
peer 192.168.2.1 multihop local-address 192.168.2.2
detect-multiplier 4
receive-interval 150
transmit-interval 150
!

```

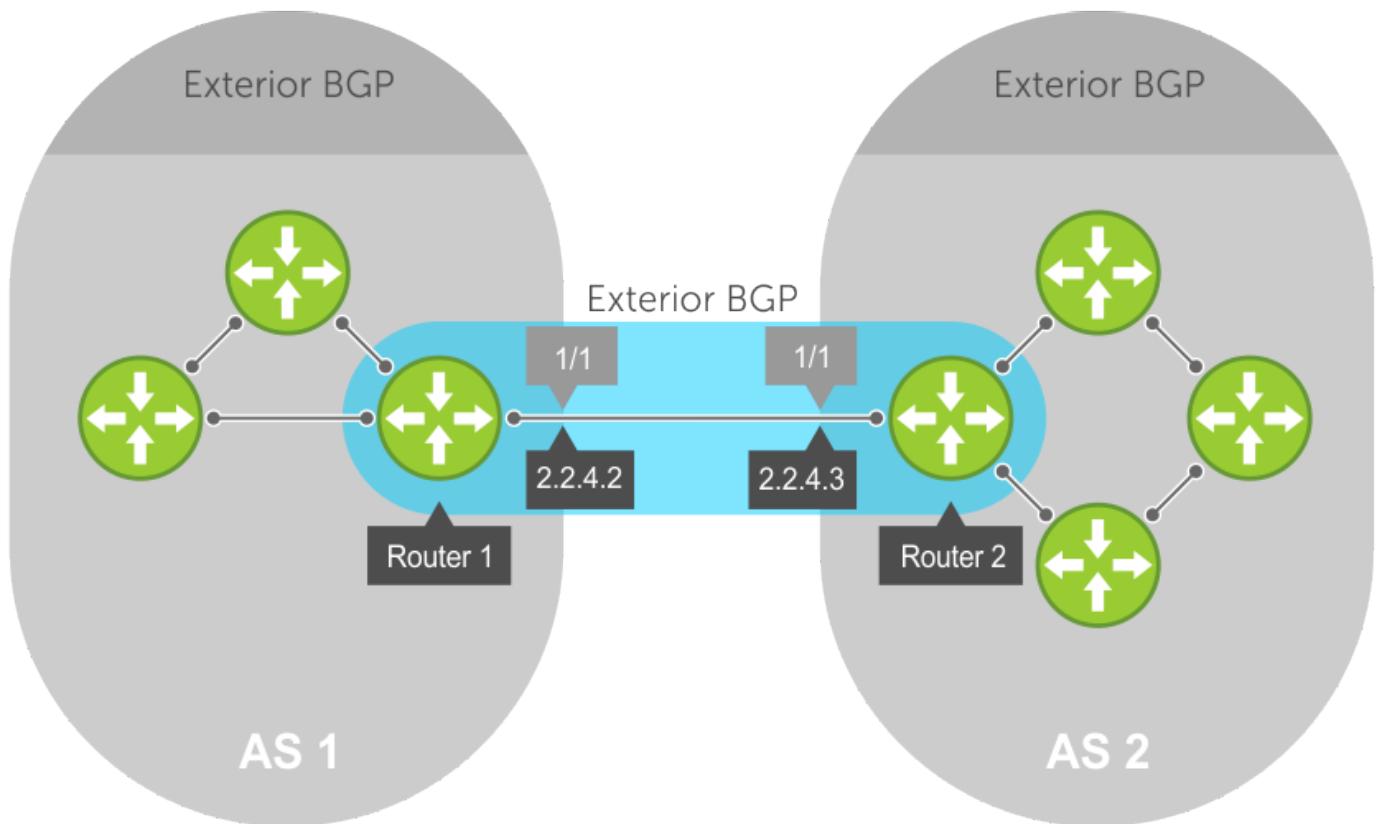
## BFD for BGP

In a BGP core network, BFD enables faster network reconvergence. BFD rapidly detects communication failures in BGP fast-forwarding paths between internal BGP (iBGP) and external BGP (eBGP) peers.

BFD for BGP is supported on physical, port channel, and VLAN interfaces. BFD for BGP does support the BGP multihop feature. Before configuring BFD for BGP, first configure BGP on the interconnecting routers.

### Example BFD to BGP

In this BFD for BGP configuration example, Router 1 and Router 2 use eBGP in a transit network to interconnect AS1 and AS2. The eBGP routers exchange information with each other and with iBGP routers to maintain connectivity and accessibility within each autonomous system.



When you configure a BFD session with a BGP neighbor, you can establish a BFD session with a specified BGP neighbor using the `neighbor ip-address` and `bfd` commands.

#### Router 1

```

sonic(config)# router bgp 1
sonic(conf-router-bgp-1)# neighbor 2.2.4.3

```

```
sonic(conf-router-bgp-neighbor) # bfd
sonic(conf-router-bgp-neighbor) # no shutdown
```

## Router 2

```
sonic(config)# router bgp 2
sonic(conf-router-bgp-2) # neighbor 2.2.4.2
sonic(conf-router-bgp-neighbor) # bfd
sonic(conf-router-bgp-neighbor) # no shutdown
```

BFD packets originating from a router are assigned to the highest priority egress queue to minimize transmission delays. Incoming BFD control packets that are received from the BGP neighbor are assigned to the highest priority queue within the control plane policing (CoPP) framework to avoid BFD packets drops due to queue congestion.

BFD notifies BGP of any failure conditions that it detects on the link. BGP initiates recovery actions. BFD for BGP is supported only on directly connected BGP neighbors and in both BGP IPv4 and IPv6 networks. A maximum of 100 simultaneous BFD sessions are supported.

If each BFD for BGP neighbor receives a BFD control packet within the configured BFD interval for failure detection, the BFD session remains up and BGP maintains its adjacencies. If a BFD for BGP neighbor does not receive a control packet within the detection interval, the router informs any clients of the BFD session, and other routing protocols, about the failure. It then depends on the routing protocol that uses the BGP link to determine the appropriate response to the failure condition. The normal response is to terminate the peering session for the routing protocol and reconverge by bypassing the failed neighboring router.

A log message generates whenever BFD detects a failure condition.

## BFD for OSPF

You can configure BFD to monitor and notify reachability status between OSPF neighbors. When you use BFD with OSPF, BFD sessions are established between all neighboring interfaces participating with OSPF full state. If a neighboring interface fails, BFD notifies OSPF protocol that a link state change has occurred.

### Configure BFD for OSPF

1. Enable BFD.
2. Configure OSPF on the interconnecting routers.

## Enable BFD

- Enable BFD.

```
sonic(conf-if-Eth1/1) # ip ospf bfd
```

## BFD for PIM

You can enable BFD support for PIM on individual interfaces.

## Enable BFD for PIM

- Enable BFD.

```
sonic(conf-if-Eth1/1) # ip pim bfd
```

# BFD profile

You can use a BFD profile to create a template of BFD configurations and apply to multiple BFD peers without configuring each BFD peer separately. BFD profile also enables changing BFD timers of dynamic sessions without configuring static BFD peers.

Within a BFD profile, you can configure all the necessary BFD parameters. When you apply the BFD profile to a static peer, BGP, OSPF, or PIM configuration, all parameters that you have configured under the profile are applied to those protocols.

## Profile configuration scenarios

1. **Scenario 1:** You can apply a BFD profile without first creating it. However, the profile takes effect only after it is configured. The default BFD settings are used until you configure the profile.
2. **Scenario 2:** A BFD profile is associated with a static BFD peer and BFD parameters are configured in the static peer as well. Parameters that are configured in the static peer take precedence over the BFD profile.
3. **Scenario 3:** BGP, OSPF, and PIM share a BFD session and the BFD profile that is associated with BGP, OSPF and PIM is different. The latest configured profile either in BGP, OSPF, or PIM takes effect.
4. **Scenario 4:** BFD profile configuration is changed dynamically. All the configuration parameters apply immediately and BFD timers are renegotiated using the polling method.
5. **Scenario 5:** BFD profile that is associated with BGP, OSPF, PIM, or BFD peer is deleted. The associated BFD session reverts to its default values. The profile configuration should be deleted from the BGP, OSPF, PIM, or BFD peer as well and reconfigured to take effect.
6. **Scenario 6:** A BFD profile is deleted. The BFD profile can be deleted from BFD without unconfiguring the profile from the protocols, if any. Similarly, the BFD profile can be unconfigured from protocols without deleting the profile in BFD. BFD profile configuration is allowed to be overwritten with the new profile without unconfiguring the existing profile.

## Configure BFD profile

- Enable BFD in CONFIGURATION mode.

```
sonic(config) # bfd
```

- Create a BFD profile in BFD CONFIGURATION mode.

```
sonic(conf-bfd) # profile profile-name
```

- After configuring the peer, configure BFD parameters. Enter the no form of a command to remove the configured BFD setting and restore the default value.

```
sonic(conf-bfd-peer) # {detect-multiplier value | echo-interval | echo-mode | minimum-ttl | passive-mode | receive-interval | transmit-interval}
```

- detect-multiplier *value* - Configure the detection multiplier to determine packet loss (2-128; default 3). The remote transmission interval is multiplied by this value to determine the connection loss detection timer. For example, if the local system has detect-multiplier 5 and the remote system has transmission interval 300 configured, the local system will detect failures only after 1500 milliseconds without receiving packets.
- echo-interval - Configure the minimum transmission interval (less jitter) that the local system uses to send BFD echo packets (10-60000; default 300).
- echo-mode - Configure echo mode.
- minimum-ttl - Configure the minimum expected TTL for incoming multihop BFD peer packets (1-254; default 254). If the TTL of the received BFD packet is less than the configured TTL, the system discards the packet.
- passive-mode - Configure a passive session in which no attempt is made to initiate connection with a BFD peer. The router waits to receive BFD control packets from the peer before the BFD session becomes active. By default, passive-mode is disabled.
- receive-interval - Configure the minimum interval during which the system can receive BFD control packets (10-60000; default 300).
- shutdown - Disable a BFD peer.
- transmit-interval - Configure the minimum transmission interval during which the system can send BFD control packets (10-60000; default 300).

## Apply a BFD profile to a static peer

You can associate a BFD profile with a BFD peer. BFD parameters that are manually configured in the static BFD peer takes precedence over this BFD profile. You can enter a maximum of 63 characters as the profile name.

- Enable BFD in CONFIGURATION mode.

```
sonic(config) # bfd
```

- Define the static BFD peer in BFD CONFIGURATION mode.

```
sonic(conf-bfd) # peer ip-address interface interface-type-number
```

- Apply the BFD profile that you configured previously to the peer.

```
sonic(conf-bfd-peer) # profile profile-name
```

### Example

```
sonic(config) # bfd
sonic(conf-bfd) # peer 10.1.1.1 interface Eth1/3
sonic(conf-bfd-peer) # profile test
```

## Apply a BFD profile to a BGP neighbor

Associate a BFD profile to a BGP neighbor. You can enter a maximum of 63 characters as the profile name.

- Configure BGP globally in CONFIGURATION mode.

```
sonic(config) # bfd
```

- Create a BFD profile.

```
sonic(conf-bfd) # profile profile-name
```

- Configure BFD settings.

```
sonic(config-bfd-profile) # detect-multiplier value
sonic(config-bfd-profile) # echo-interval value
sonic(config-bfd-profile) # echo-mode
sonic(config-bfd-profile) # receive-interval value
sonic(config-bfd-profile) # transmit-interval value
```

- Apply the BFD profile to the BGP neighbor or peer group.

```
sonic(config) # router bgp local_asn
sonic(conf-router-bgp) # neighbor {ip-address | ipv6-address}
sonic(conf-router-bgp-neighbor) # bfd profile profile-name
```

Or

```
sonic(config) # router bgp local_asn
sonic(conf-router-bgp) # neighbor {ip-address | ipv6-address}
sonic(conf-router-bgp-neighbor) # peer-group peer-group-name
sonic(conf-router-bgp-pg) # bfd profile profile-name
```

### Example

```
sonic(config) # router bgp 1
sonic(config-router-bgp) # neighbor 10.1.1.1
sonic(config-router-bgp-neighbor) # bfd profile test
```

```
sonic(config) # router bgp 1
sonic(config-router-bgp) # peer-group test
sonic(config-router-bgp-pg) # bfd profile test
sonic(conf-bfd) # profile profile-cx-1
sonic(config-bfd-profile) # detect-multiplier 5
```

```
sonic(config-bfd-profile)# echo-interval 200
sonic(config-bfd-profile)# echo-mode
sonic(config-bfd-profile)# receive-interval 200
sonic(config-bfd-profile)# transmit-interval 200
```

```
sonic(config)# router bgp 500
sonic(config-router-bgp)# neighbor 10.10.150.2
sonic(config-router-bgp-neighbor)# bfd profile profile-cx-1
```

Or

```
sonic(config)# router bgp 500
sonic(config-router-bgp)# neighbor 10.10.150.2
sonic(config-router-bgp-neighbor)# peer-group bgp-cx-1
sonic(config-router-bgp-pg)# bfd profile profile-cx-1
```

## Apply a BFD profile to an OSPF-enabled interface

Associate a BFD profile to an OSPF-enabled interface. You can enter a maximum of 63 characters as the profile name.

- Enter the interface mode.

```
sonic(config)# interface interface-type-number
```

- Apply the BFD profile that you configured previously on the interface.

```
sonic(conf-if-Eth1/3)# ip ospf bfd profile profile-name
```

### Example

```
sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# ip ospf bfd profile test
```

## Apply a BFD profile to PIM

Associate a BFD profile to an PIM-enabled interface. You can enter a maximum of 63 characters as the profile name.

- Enter the interface mode.

```
sonic(config)# interface interface-type-number
```

- Apply the BFD profile that you configured previously on the interface.

```
sonic(conf-if-Eth1/3)# ip pim bfd profile profile-name
```

### Example

```
sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# ip pim bfd profile test
```

## Configure BFD passive mode

You can configure the BFD passive mode with or without a BFD profile.

- Configure BFD passive mode without a BFD profile.
  - Define the static BFD peer in BFD CONFIGURATION mode.

```
sonic(conf-bfd)# peer ip-address interface interface-type-number
```

- Configure passive mode.

```
sonic(conf-bfd-peer)# passive-mode
```

- Configure BFD passive mode with a BFD profile.
  - Enter a BFD profile.

```
sonic(conf-bfd) # profile profile-name
```

- Configure passive mode.

```
sonic(conf-bfd-profile) # passive-mode
```

### Example

```
sonic(config) # bfd
sonic(conf-bfd) # peer 10.1.1.1 interface Eth1/3
sonic(conf-bfd-peer) # passive-mode
```

```
sonic(config) # bfd
sonic(conf-bfd) # profile test
sonic(conf-bfd-profile) # passive-mode
```

## Configure the minimum TTL for a multihop peer

Configure the minimum TTL for a multihop BFD peer. The default TTL value for a multihop peer is 254.

- Configure the minimum TTL without a BFD profile.
  - Define the static BFD peer in BFD CONFIGURATION mode.

```
sonic(conf-bfd) # peer ip-address local-address local-ip-address multihop
```

- Configure the minimum TTL value (1 to 254).

```
sonic(conf-bfd-peer) # minimum-ttl ttl-value
```

- (Optional) Configure the minimum TTL without a BFD profile.
  - Enter a BFD profile.

```
sonic(conf-bfd) # profile profile-name
```

- Configure the minimum TTL value (1 - 254).

```
sonic(conf-bfd-profile) # minimum-ttl ttl-value
```

### Example

```
sonic(config) # bfd
sonic(conf-bfd) # peer 10.1.1.1 local-address 10.1.1.2 multihop
sonic(conf-bfd-peer) # minimum-ttl 253
```

```
sonic(config) # bfd
sonic(conf-bfd) # profile test
sonic(conf-bfd-profile) # minimum-ttl 253
```

## View BFD profile

- Use the following command to view the BFD profile that is configured on the system:

```
sonic# show bfd profile [profile-name]
```

### Example

View all BFD profiles that are configured.

```
sonic# show bfd profile
BFD Profile:
```

```
Profile-name: fast
 Enabled: True
 Echo-mode: Enabled
 Passive mode: Enabled
 Minimum-Ttl: 254
 Detect-multiplier: 4
 Receive interval: 123ms
 Transmission interval: 123ms
 Echo transmission interval: 1234ms

BFD Profile:
 Profile-name: test
 Enabled: True
 Echo-mode: Enabled
 Passive mode: Enabled
 Minimum-Ttl: 254
 Detect-multiplier: 4
 Receive interval: 123ms
 Transmission interval: 123ms
 Echo transmission interval: 1234ms
```

View a specific BFD profile.

```
sonic# show bfd profile fast
BFD Profile:
 Profile-name: fast
 Enabled: True
 Echo-mode: Enabled
 Passive mode: Enabled
 Minimum-Ttl: 254
 Detect-multiplier: 4
 Receive interval: 123ms
 Transmission interval: 123ms
 Echo transmission interval: 1234ms
```

## View BFD peer information

View BFD peer information.

```
sonic# show bfd peers
BFD Peers:
 peer 172.11.0.1 vrf default interface Vlan101
 ID: 2604839737
 Remote ID: 2286829245
 Passive mode: Disabled
 Profile: bfd_prof_0
 Status: up
 Uptime: 0 day(s), 23 hour(s), 8 min(s), 14 sec(s)
 Diagnostics: ok
 Remote diagnostics: ok
 Peer Type: dynamic
 Local timers:
 Detect-multiplier: 3
 Receive interval: 300ms
 Transmission interval: 300ms
 Echo transmission interval: 0ms
 Remote timers:
 Detect-multiplier: 3
 Receive interval: 300ms
 Transmission interval: 300ms
 Echo transmission interval: 300ms
```

View multihop BFD peer information.

```
sonic# show bfd peer 10.1.1.2 multihop local-address 10.1.1.1 vrf default
peer 10.1.1.2 multihop local-address 10.1.1.1 vrf default
 ID: 82748345
 Remote ID: 0
 Active mode
```

```

Minimum TTL: 123
Status: down
Downtime: 0 day(s), 0 hour(s), 0 min(s), 19 sec(s)
Diagnostics: ok
Remote diagnostics: ok
Peer Type: configured
Local timers:
 Detect-multiplier: 3
 Receive interval: 300ms
 Transmission interval: 300ms
 Echo transmission interval: 60ms
Remote timers:
 Detect-multiplier: 3
 Receive interval: 1000ms
 Transmission interval: 1000ms
 Echo transmission interval: 0ms

```

View single hop BFD peer information

```

sonic# show bfd peer 10.1.1.2 vrf default interface Eth1/3

peer 10.1.1.2 vrf default interface Eth1/3
 ID: 2286155092
 Remote ID: 0
 Passive mode
 Status: down
 Downtime: 0 day(s), 0 hour(s), 1 min(s), 6 sec(s)
 Diagnostics: ok
 Remote diagnostics: ok
 Peer Type: configured
 Local timers:
 Detect-multiplier: 3
 Receive interval: 300ms
 Transmission interval: 300ms
 Echo transmission interval: 0ms
 Remote timers:
 Detect-multiplier: 3
 Receive interval: 1000ms
 Transmission interval: 1000ms
 Echo transmission interval: 0ms
sonic# show bfd peer 172.11.0.1 vrf default interface Vlan 101
BFD Peers:
 peer 172.11.0.1 vrf default interface Vlan101
 ID: 2604839737
 Remote ID: 2286829245
 Passive mode: Disabled
 Profile: bfd_prof_0
 Status: up
 Uptime: 0 day(s), 23 hour(s), 17 min(s), 26 sec(s)
 Diagnostics: ok
 Remote diagnostics: ok
 Peer Type: dynamic
 Local timers:
 Detect-multiplier: 3
 Receive interval: 300ms
 Transmission interval: 300ms
 Echo transmission interval: 0ms
 Remote timers:
 Detect-multiplier: 3
 Receive interval: 300ms
 Transmission interval: 300ms
 Echo transmission interval: 300ms

```

# Link state tracking

Link state tracking indicates the loss of upstream connectivity for downstream servers connected to the switch. A switch provides connectivity for devices, such as servers.

If the switch loses upstream connectivity, the downstream devices also lose connectivity. However, the downstream devices do not generally receive an indication that the upstream connectivity was lost because connectivity to the switch is still operational.

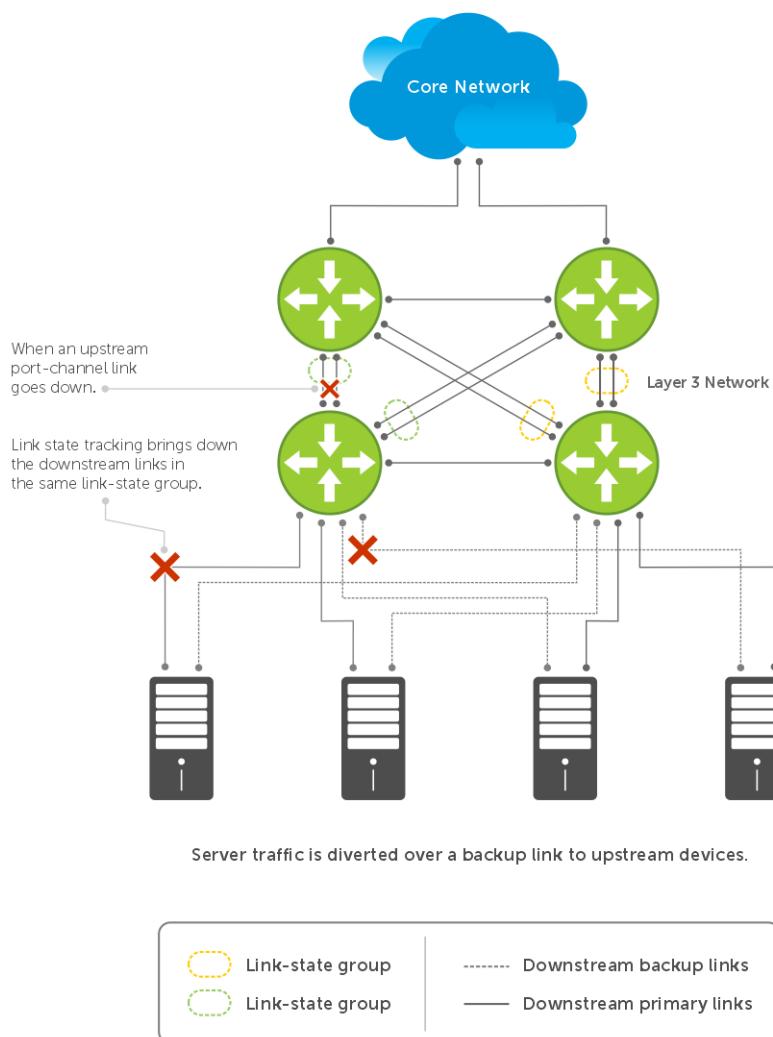
Link state tracking associates downstream interfaces with upstream interfaces. When upstream connectivity fails, the switch operationally disables its downstream links. Failures on the downstream links allow downstream devices to recognize the loss of upstream connectivity. This allows the downstream servers to select alternate paths, if available, to send traffic to upstream devices.

Link state tracking creates an association between upstream and downstream interfaces in a *link-state group*. An interface in a link-state group can be a physical Ethernet, fiber channel interface, or a port channel interface. An interface in a link-state group can also be a VLAN interface as an upstream interface.

By default, if all the upstream interfaces in a link-state group go down, all the downstream interfaces in the same link-state group are set into a Link-Down state. In addition, in a link-state group, you can configure automatic recovery of downstream ports when there is a change in the link status of uplink interfaces. You can also bring up downstream interfaces that are in an *iftrack-status-down* state manually.

 **NOTE:** A link state group can consist of both upstream and downstream interfaces.

## Link state tracking example



## Link state tracking configuration notes

- An link-state group is considered to be operationally up if it has at least one upstream interface in the Link-Up state.
- An link-state group is considered to be operationally down if it has no upstream interfaces in the Link-Up state .
- Any physical port, port channel interface, or VLAN interface can be assigned as an uplink interface to a link-state group. Uplink interfaces that are assigned to a link-state group must be Layer 3 interfaces, assigned with an IP address. You cannot assign a VLAN interface as a downstream interface to a link-state group.
- You can assign an interface to only one link-state group at a time .
- You can designate the link-state group as either an upstream or downstream interface, but not both.
- You cannot assign both a port channel and its members to link-state group, which would make the group inactive. The port channels and individual ports that are not part of any port channel can coexist as members of a link-state group.
- If one of the upstream interfaces in a link-state group goes down, you can set the downstream ports in an operationally down state with an `iftrack-status-down` error status. You can configure the system to disable either a user-configurable set of downstream ports or all the downstream ports in the group.
- When an upstream interface in a link-state group that was down comes up, the set of disabled downstream ports that were down due to that particular upstream interface are brought up, and the `iftrack-status-down` error clears in those downstream ports.
- If you disable a link-state group, the downstream interfaces are not disabled, regardless of the state of the upstream interfaces. If you do not assign upstream interfaces to a link-state group, the downstream interfaces are not disabled.
- You can configure link state tracking on an MLAG device. All downstream interfaces of a link-state group are operationally down until the configured delay-restore timer expires. You configure the delay-restore timer in an MLAG domain.

### Example: Link state tracking on an MLAG device

Use link state tracking to allow traffic from downstream links to be sent to an MLAG peer if all upstream links on the local switch are down — see [MLAG configuration](#). Downstream MLAG interfaces on the local switch are shut down. Upstream links on the MLAG peer transmit traffic to the spine. Link state tracking avoids the need to increase bandwidth on the peer link to handle the additional upstream traffic. For example:

```
sonic(config)# link state track trackPeer1
sonic(config-link-track)# downstream all-mlag
sonic(config-link-track)# exit
```

### Example: Link state tracking on an EVPN multihomed VTEP

Configure uplink tracking to avoid black-holing traffic from a tenant device in case an EVPN multihomed VTEP gets isolated from the network — see [EVPN multihoming](#). The multihomed downlink interfaces on the VTEP are shut down if all uplink interfaces on the VTEP go down. You can specify an optional timeout value (in seconds) to wait before bringing up a multihomed interface after one or more uplink interfaces come up. Create a track group and associate all EVPN multihoming interfaces to the track group; for example:

```
sonic(config)# link state track trackGrp
sonic(config-link-track)# downstream all-evpn-es
sonic(config-link-track)# timeout 300
sonic(config-link-track)# exit
sonic(config)# interface Ethernet0
sonic(config-if-Ethernet0)# link state track trackGrp upstream
```

## Unidirectional link detection

Unidirectional link detection (UDLD) is a Layer 2 protocol for the detection of unidirectional link failures, which is required in spanning-tree topologies to avoid loops.

All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it disables the port.

## Operational modes

UDLD supports two operational modes which are user-configurable:

**Normal mode** In normal mode, the failure detection is always *event based*. The event in this case is the reception of PDU. All actions are taken based on the information learned from the received PDU. In case UDLD stops receiving the PDU, then there is no action that is taken (like shutting down the interface, however a log message is generated in this case), this is a conservative approach that is taken to minimize the false positives during failure detection process. In case the received information from the PDU indicates failure then action is taken to shut down the interface.

**Aggressive mode** Aggressive mode behavior is similar to normal mode, except the case when the link is in bi-directional state and then stops receiving the PDUs. This is treated as a meaningful network event and UDLD takes the action of shutting down the interface.

**i | NOTE:** By default, UDLD is enabled in normal mode.

In aggressive mode, UDLD detects a unidirectional link by using the previous detection methods. UDLD in aggressive mode can also detect a unidirectional link on a point-to-point link on which no failure between the two devices is allowed. It can also detect a unidirectional link when one of these problems exists:

- One of the ports cannot send or receive traffic
- One of the ports is down while the other is up
- One of the fiber strands in the cable is disconnected

In these cases, UDLD disables the affected port.

In a point-to-point link, UDLD hello packets can be considered as a heart beat whose presence guarantees the health of the link. The loss of the heart beat means that the link must be shut down if it is not possible to reestablish a bi-directional link.

If both fiber strands in a cable are working from a Layer 1 perspective, UDLD in aggressive mode detects whether those fiber strands are connected correctly and if the traffic is flowing bi-directionally between the correct neighbors. This check cannot be performed by autonegotiation because autonegotiation operates at Layer 1.

## Failure detection

**Empty echo** UDLD protocol data unit (PDU) is transmitted on the port but, when received by the neighbor, PDU does not contain the local device information (no echo) resulting in an empty echo state. On detecting this condition, the interface is shut down to both aggressive and normal mode.

**Incoming UDLD packets stop** When the link is in bi-directional state and the UDLD PDU reception stops, this results in the detection of unidirectional failure after the timeout period. On detecting this condition, the interface is shut down for aggressive mode and the UDLD state is set as undetermined for normal mode.

**Tx-Rx loop** When the UDLD PDU transmitted on the port is received back it is detected as Tx-Rx loop, and the interface is shut down. This behavior is the same for both aggressive and normal mode.

## Port states

**Undetermined** UDLD has not received enough information from the peer to determine the state.

**Shutdown** UDLD has determined that the link is unidirectional and has shutdown the interface.

**Bi-directional** UDLD can exchange the PDUs and has determined the link to be bi-directional.

## Failure recovery

Once UDLD has detected a failure and has shutdown the interface, you can recover the interface with `shutdown` or `no shutdown`.

After recovery when the port comes up, UDLD exchanges the PDUs with the neighboring device to determine the link state. During this period, the kernel interface continues to be in the down state. For all system applications, the port is considered to be down. Once UDLD determines the port is bi-directional, the kernel interface state is moved to up.

The kernel interface state can be controlled by user space applications using the interface-specific flags available. During failure condition, UDLD sets the `IF_OPER_DORMANT` flag and then trigger the port shutdown. After recovery when the port

comes up, kernel will not move the port state to IF\_OPER\_UP as IF\_OPER\_DORMANT flag is set, kernel waits for user space application to update the oper state. Once UDLD determines the link is bi-directional, the port state is updated with IF\_OPER\_UP. This results in net link notifications to all the subscribed applications that the port is operationally up now. This mechanism ensures that the port continues to be operationally down when the failure recovery is in progress and avoids the unnecessary port state change if the failure condition continues to persist.

## LAG member port

UDLD can be enabled on the member ports of port channel interfaces. Operationally the UDLD behavior on the member ports is the same as individual Ethernet ports. When UDLD detects a failure and shuts down the port, the kernel state for the port is moved to down.

## Configure UDLD

You can enable UDLD globally in normal or aggressive mode, or you can enable UDLD on a specific interface. UDLD is disabled by default. Individual interface configuration overrides UDLD global configuration.

### Configuration considerations

- Both ends of the links must be UDLD enabled for the protocol to detect the unidirectional failures
- UDLD mode (aggressive or normal) should be the same on both ends of the links. Different modes on both sides can result in delayed detection of unidirectional failures.
- UDLD aggressive mode should only be used on point-to-point links
- Autonegotiation and link fault signaling (LFS) operate on Layer 1 and can discover link issues, such as unidirectional failures, and bring the link down. UDLD is still useful in such cases when the link is up at Layer 1 but still has unidirectional failures due to miswiring or for any other reason.

### Enable UDLD globally

1. Enable UDLD.

```
sonic(config) # udld enable
```

2. Specify the UDLD mode of operation.

```
sonic(config) # udld aggressive
```

3. Specify the UDLD message time interval (1 to 30 seconds; default 1).

```
sonic(config) # udld message-time msg-time
```

4. Specify the UDLD multiplier value (3 to 10 seconds; default 3).

```
sonic(config) # udld multiplier multiplier
```

### Enable UDLD on an interface

1. Enter INTERFACE mode.

```
sonic(config) # interface Eth slot/port [/breakout-port]
```

2. Enable UDLD.

```
sonic(conf-if-Eth) # udld enable
```

3. Specify the UDLD mode of operation.

```
sonic(conf-if-Eth) # udld aggressive
```

- Specify the UDLD message time interval (1 to 30 seconds; default 1).

```
sonic(conf-if-Eth)# udld message-time msg-time
```

- Specify the UDLD multiplier value (3 to 10 seconds; default 3).

```
sonic(conf-if-Eth)# udld multiplier multiplier
```

## View UDLD configuration

You can use these show commands to view UDLD information.

### View UDLD global configuration

```
sonic# show udld global
UDLD Global Information
 Admin State : UDLD Enabled
 Mode : Normal
 UDLD Message Time : 2 seconds
 UDLD Multiplier : 4
```

### View UDLD interface configuration

```
sonic# show udld interface Eth1/2
UDLD information for Eth1/2
 UDLD Admin State: Enabled
 Mode: Normal
 Status: Bidirectional
 Local device id: 3c2c.992d.8201
 Local port id: Eth1/2
 Local device name: Sonic
 Message time: 2
 Timeout interval: 4
 Neighbor Entry 1

 Neighbor device id: 3c2c.992d.8235
 Neighbor port id: Eth1/2
 Neighbor device name: Sonic
 Neighbor message time: 1
 Neighbor timeout interval: 3
```

### View UDLD neighbors

```
sonic# show udld neighbors
Port Device Name Device ID Port ID Neighbor State
-----+-----+-----+-----+-----+-----+
Eth1/2 Sonic 3c2c.992d.8201 Eth1/4 Bidirectional
Eth1/3 Sonic 3c2c.992d.8201 Eth1/5 Bidirectional
```

### View UDLD statistics

```
sonic# show udld statistics
UDLD Interface statistics for Eth1/2
 Frames transmitted: 10
 Frames received: 9
 Frames with error: 0

UDLD Interface statistics for Eth1/3
 Frames transmitted: 5
 Frames received: 8
 Frames with error: 0
```

### View UDLD statistics by interface

```
sonic# show udld statistics interface Eth1/2
UDLD Interface statistics for Eth1/2
 Frames transmitted: 10
```

|                    |   |
|--------------------|---|
| Frames received:   | 9 |
| Frames with error: | 0 |

## Link-error port disabling

Link-error port disabling prevents excessive interface flapping events from adversely affecting routing protocols and routing tables in the network. By suppressing port state-change events, you protect system resources.

A link flap occurs when a port interface repeatedly goes up and down. To minimize disruption to the switch and network operation, you can enable link-error disabling on a port interface for a specified recovery time.

When enabled, link-error disabling records when a port state changes from up to down for a specified number of times during a specified time period before the interface is physically disabled for a specified recovery wait period. The following settings are used to suppress port link-state events and protect system resources:

- flap threshold — Number of times that a port link-state goes from up to down before the recovery wait period is activated.
- sampling interval — Time (in seconds) during which the flap threshold can occur before the recovery wait period is activated.
- recovery interval — Time (in seconds) during which the port remains disabled (down) before it is re-enabled (up). Enter 0 seconds to keep a port link-state disabled until it is manually re-enabled or until link-error disabling is unconfigured on the port.

By default, link-error disabling is disabled on physical ports on a switch. You must enable link-error disabling globally and on specified interfaces. If you save link-error disabling configuration settings, they are maintained across switch reboots.

### Configure link-error disabling

1. Configure link-error disabling on specified interfaces, including interface ranges. The valid values are:

- flap threshold (number of times): 1 to 50; default 3.
- sampling interval (in seconds): 1 to 65535; default 10.
- recovery interval (in seconds): 0 to 65534; default 300.

```
sonic(conf-if-Ethslot/port)# link-error-disable [flap-threshold number] [sampling-interval seconds] [recovery-interval seconds]
```

For example:

```
sonic(config)# interface Eth1/4
sonic(conf-if-Eth1/4)# link-error-disable flap-threshold 10 sampling-interval 3
recovery-interval 10
```

```
sonic(config)# interface range Eth 1/1-1/2,1/7,1/12-1/15
sonic(conf-if-range-eth**)# link-error-disable flap-threshold 10 sampling-interval 3
recovery-interval 10
```

To configure link-error disabling using the default values of flap threshold, sampling internal, and recovery interval, enter the `link-error-disable` command; for example:

```
sonic(conf-if-Eth1/4)# link-error-disable
```

```
sonic(conf-if-range-eth**)# link-error-disable
```

To unconfigure link-error disabling on specified interfaces, enter the `no link-error-disable` command; for example:

```
sonic(conf-if-Eth1/4)# no link-error-disable
```

```
sonic(conf-if-range-eth**)# no link-error-disable
```

2. Enable link-error disabling globally on the switch. Link-error disabling is enabled only on the port interfaces that you configured with an interface-specific `link-error-disable` configuration in Step 1.

```
sonic(config)# errdisable recovery cause link-flap
```

To disable link-error disabling on all switch interfaces, enter the `no errdisable recovery cause link-flap` command.

### View link-error disabling

To view the global status of link-error disabling and the recovery wait period, use the `show errdisable recovery` commands.

```
sonic# show errdisable recovery
Err-Disable Reason Status

udld Disabled
bpduguard Disabled
link-flap Enabled
Timeout for Auto-recovery: 300 seconds
```

To view the link-error disabling configuration and status on individual port interfaces, use the `show errdisable link-flap` command. Interfaces which are not configured with specific link-error-disable settings are not displayed.

```
sonic# show errdisable link-flap
Interface Flap-threshold Sampling-interval Recovery-interval Time-left Status

Management0 10 3 30 23 Err-disabled
Eth1/4 10 3 60 N/A On
Eth1/8 5 10 300 N/A Off
```

The possible link-error-disable statuses are:

- **Err-disabled:** The number of link flaps in a sampling interval exceeded the threshold; link-error disabling is disabled on the port (err-disabled state).
- **Off:** Link-error disabling settings are configured, but the feature is not enabled.
- **On:** Link-error disabling is enabled; no link flaps have been detected.

# sFlow

sFlow monitors network traffic in high-speed networks with many switches and routers. The collected data on inbound and outbound traffic is sent to an sFlow data collector.

- SONiC supports sFlow version 5.
- sFlow data collection is only supported on data ports.
- You can configure a maximum of two sFlow collectors.

sFlow uses two types of sampling:

- Statistical packet-based sampling of switched or routed packet flows
- Time-based sampling of interface counters

sFlow monitoring consists of an sFlow agent that is embedded in a switch and an sFlow collector:

- The sFlow agent resides anywhere within the path of the packet. The agent combines the flow samples and interface counters into sFlow datagrams and forwards them to the sFlow collector at regular intervals. The datagrams consist of information about, but not limited to, the packet header, ingress and egress interfaces, sampling parameters, and interface counters. ASICs handle packet sampling.
- The sFlow collector analyses the datagrams that are received from different devices and produces a network-wide view of traffic flows.

## Topics:

- [Configure sFlow](#)
- [View sFlow statistics](#)
- [sFlow configuration example](#)

## Configure sFlow

Configure sFlow globally on a switch. By default, the sFlow agent is disabled. You must enable sFlow globally to sample traffic on all data interfaces before you can reconfigure the default settings.

Reconfigure the sFlow sampling rate for packets only in exceptional cases. The sampling rate collects one packet in the specified number of packets (256 to 8388608). For example, if you configure a sampling rate of 256, the system samples one packet out of 256 packets. The default detects a new flow of 10% of the link bandwidth in less than one second and depends on the interface speed — see **sFlow defaults**. It is recommended that you do not change the default setting.

You can configure the sampling globally, on an interface, or a range of interfaces. When you configure the sampling rate for an interface or a range of interfaces, the value takes precedence over the global configuration. When you undo the global configuration using the `no sflow sampling-rate` command, the sampling rate of all interfaces that do not have an interface-level sampling rate configuration is reset to the default value.

### sFlow defaults

- sFlow polling interval — 20 seconds
- sFlow collector port — 6343
- sFlow collector VRF — Default VRF
- sFlow sampling rates:
  - 1G link — 1 packet in 1000
  - 10G link — 1 packet in 10,000
  - 40G link — 1 packet in 40,000
  - 50G link — 1 packet in 50,000
  - 100G link — 1 packet in 100,000

### sFlow configuration

1. Enable sFlow globally on all inbound and outbound interfaces.

```
sonic(config) # sflow enable
```

You can disable sFlow on a per-interface basis in Interface Configuration mode.

```
sonic(config) # interface Eth1/2
sonic(conf-if-Eth1/2) # no sflow enable
```

2. Configure an sFlow collector by entering its IPv4 or IPv6 address. Configure the destination collector-port number for sFlow data traffic (0 to 65535; default 6343). Specify the VRF in which the sFlow collector operates: Management (mgmt) or default. The default VRF on the collector is used by default. You can configure the same collector IP address and/or port number in different VRFs.

```
sonic(config) # sflow collector {ip-address | ipv6-address} [collector-port-number]
[vrf vrf-name]
```

```
sonic(config) # sflow collector 1.1.1.2 4451 vrf mgmt
sonic(config) # sflow collector 1.1.1.2 4451 vrf default
```

To remove an sFlow collector, enter the no version of the command:

```
no sflow collector {ip-address | ipv6-address} [collector-port-number] [vrf vrf-name]
```

3. Configure a non-default sFlow polling interval and maintain the default sampling rate. The polling interval is the time (in seconds) when traffic samples or counters are collected (5 to 300; default 20). Enter 0 to disable sFlow traffic polling.

```
sonic(config) # sflow polling-interval seconds
```

```
sonic(config) # sflow polling-interval 44
```

To return the sFlow polling interval to the default, enter no sflow polling-interval.

4. Configure an sFlow agent interface. The interface name provides the IPv4 or IPv6 address for the collector to uniquely identify the source of the packets it receives.

 **NOTE:** Dell Technologies recommends that you configure an sFlow agent interface.

```
sonic(config) # sflow agent-id interface-name
```

```
sonic(config) # sflow agent-id Eth1/2
```

To return the sFlow agent interface to the default, enter no sflow agent-id.

5. (Optional) To configure the sampling rate globally:

```
sonic(config) # sflow sampling-rate sampling-rate
```

```
sonic(config) # sflow sampling-rate 4400
```

To configure the sampling rate for an interface:

```
sonic(conf-if) # sflow sampling-rate sampling-rate
```

```
sonic(conf-if-Eth1/2) # sflow sampling-rate 4400
```

# View sFlow statistics

Use the show commands to view sFlow configuration and counters.

## View global sFlow configuration

```
sonic# show sflow

Global sFlow Information

admin state: down
polling-interval: default
agent-id: default
sampling-rate: 256
configured collectors: 0
```

## View sFlow interface status

```
sonic# show sflow interface

sFlow interface configurations
 Interface Admin State Sampling Rate
 Eth1/2 up 4000
 Eth1/3 up 4000
 Eth1/4 up 4000
 Eth1/5 up 4000
 Eth1/6 up 4000
 Eth1/7 up 4000
 Eth1/8 up 4000
 Eth1/9 up 4000
 Eth1/10 up 4000
 ...
 Eth1/26 up 4000
 Eth1/27 up 4000
 Eth1/28 up 4000
 Eth1/29 up 4000
 Eth1/30 up 4000
 Eth1/31 up 4000
```

# sFlow configuration example

sFlow provides a flow-based sampling method to monitor network traffic. Use sFlow to monitor network security in large enterprise data centers, monitor traffic for different tenants in a logical network and on specified interfaces, and for Quality of Service (QoS) operations.

In this example, an sFlow collector has IP address 190.167.1.1/24 with these configuration steps:

1. Enable sFlow globally on all interfaces.
2. Configure the sFlow collector.
3. Configure nondefault polling interval. Keep the default sampling rate.
4. Configure an sFlow agent interface.
5. Verify the sFlow configuration.

```
sonic(config)# sflow enable
sonic(config)# sflow collector 190.167.1.1 6343
sonic(config)# sflow polling-interval 44
sonic(config)# sflow agent-id Management 0
sonic(config)# exit

sonic # show sflow

Global sFlow Information

admin state: up
polling-interval: 44
agent-id: Management0
configured collectors: 1
```

```
190.167.1.1 6343 default

sonic# show sflow interface

sFlow interface configurations
 Interface Admin State Sampling Rate
 Eth1/2 up 4000
 Eth1/3 up 4000
 Eth1/4 up 4000
 Eth1/5 up 4000
 Eth1/6 up 4000
 Eth1/7 up 4000
 Eth1/8 up 4000
 ...
 ...
```

To disable sFlow on an interface, use the `no sflow enable` command in Interface configuration mode.

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# no sflow enable
sonic(conf-if-Eth1/2)# exit
sonic(config)# interface Eth1/3
sonic(conf-if-Eth1/3)# no sflow enable
sonic(conf-if-Eth1/3)# exit
```

# REST API

**i | NOTE:** The REST API is supported in the Cloud Standard, Cloud Premium, Enterprise Standard, and Enterprise Premium bundles. The REST API is supported in the Edge Standard bundle only on E-series switches.

The REST API allows applications and scripts to have complete, model-driven programmatic control over the SONiC operating system, using a standard data model and easy-to-use syntax. REST API requests use uniform resource identifiers (URIs) to define the switch resources to be configured or retrieved. The REST API is enabled by default and complies with RFC 8040.

To view REST API end-points and operations, use a web browser to access the REST API Explorer page at `https://your-switch-ip-address/ui`; for example, `https://10.42.0.69/ui`. All available YANG and SONiC data models are displayed. Use REST API client tools, such as Swagger or Postman, to generate code from these data models for HTTP requests.

In addition, for a complete list of Enterprise SONiC YANG models, go to the [Dell Technologies Support](#) site, search for Enterprise SONiC Distribution, open the Drivers & Downloads page, select a software release, and download the **Enterprise SONiC OS datamodel** zip file.

## Read and write operations in HTTP requests

To configure and monitor a switch, use REST API client tools, such as Postman or Swagger, to run HTTP web requests — GET, PUT, POST, DELETE, and PATCH. These REST API operations act on SONiC REST resources:

- Configuration and operational data that the REST API client accesses in `/restconf/data`.
- Protocol-specific data model operations in `/restconf/operations` that SONiC advertises.

The REST API supports the following HTTP requests for read and write operations. REST API operations are performed on a switch resource that is identified by a Universal Resource Identifier (URI). Examples of curl commands that access a switch using the username admin and the password adminpw for the URI of a collection and a member resource are:

```
curl -k -u admin:adminpw https://switch-ip-address/restconf/data/openconfig-
```

```
interfaces:interfaces/
```

```
curl -k -u admin:adminpw https://switch-ip-address/restconf/data/openconfig-
```

```
interfaces:interfaces/interface=Eth1%2F2
```

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GET</b>    | On a collection resource — retrieves the URIs of member resources in the collection resource in the response body.<br><br>On a member resource — retrieves the representation of the member resource in the response body.                                                                                                                                                                                                                                                                                    |
| <b>POST</b>   | On a collection resource — creates a member resource in the collection resource using the instructions in the request body. The URI of the created member resource is automatically assigned and returned in the Location header field in the response.<br><br>On a member resource: Creates a member resource in the member resource using the instructions in the request body. The URI of the created member resource is automatically assigned and returned in the Location header field in the response. |
| <b>PUT</b>    | On a collection resource — replaces all representations of the member resources in the collection resource with the representation in the request body, or creates the collection resource if it does not exist.<br><br>On a member resource: Replaces all representations of the member resource or creates the member resource if it does not exist, with the representation in the request body.                                                                                                           |
| <b>PATCH</b>  | On a collection resource — updates all representations of the member resources in the collection resource using the instructions in the request body, or creates the collection resource if it does not exist.<br><br>On a member resource: Updates all representations of the member resource, or creates the member resource if it does not exist, using the instructions in the request body.                                                                                                              |
| <b>DELETE</b> | On a collection resource — deletes all representations of the member in the collection resource.                                                                                                                                                                                                                                                                                                                                                                                                              |

On a member resource: Deletes all representations of the member resource.

**i** **NOTE:** The GET method is a safe, read-only operation — applying it to a resource does not result in a state change of the resource. The GET, PUT and DELETE methods are idempotent — applying them multiple times to a resource results in the same state change of the resource as applying them once, although the response may differ.

## REST error codes

REST supports only standard HTTP error codes.

### Topics:

- REST API authentication
- View REST API authentication
- REST API requests using curl
- YANG PATCH operation
- REST API examples

## REST API authentication

For user authentication, the REST API uses HTTP basic password authentication, client certificates, and JSON Web Token (JWT)-coded tokens with `username` and `password` credentials to authenticate requests. User credentials are sent as an HTTP Authorization header in Base64 format; for example: "Authorization: Basic YWRtaW46c29uaWNhZG1pbg==".

By default, HTTP password and JWT are enabled for REST API authentication on a switch. To verify the currently enabled REST API authentication modes:

```
sonic# show ip rest authentication

REST Client Authentication Modes

client_auth: password,jwt
```

To reconfigure the REST API authentication modes:

```
sonic(config)# ip rest authentication auth-mode
```

Where `auth-mode` is one or more of the following values that are separated by commas:

- `password` — Enable HTTP password authentication.
- `jwt` — Enable JWT token-based authentication.
- `cert` — Enable client certificate authentication.
- `none` — Remove the configured authentication modes, and restore the defaults: HTTP password and JWT authentication.

Enter multiple values for `auth-mode` by separating them with a comma; for example:

```
sonic(config)# ip rest authentication password,jwt,cert
```

To view the settings used in REST API authentication, use the `show ip rest` command.

```
sonic# show ip rest
Log level is not-set
Port is 443
Request limit is 0
Read timeout is 30 seconds
Client authentication mode is password,jwt
Security profile is mysp
API timeout is 900 seconds
```

To reconfigure the settings used in REST API authentication, use the `ip rest` command.

```
sonic(config)# ip rest {log-level severity-level | port Ethslot/port | read-timeout
seconds | request-limit number | security-profile profile-name}
```

- `log-level severity-level` — Enter the severity level of messages to be logged for debugging (1-7), where lower numbers indicate more severe conditions: alerts for immediate action (1), critical conditions (2), errors (3), warnings (4), notifications (5), informational (6), debugging (7); no default.
- `port number` — Enter the TCP port number used by the REST server to receive REST requests (1-65535; default 443).
- `read-timeout seconds` — Enter the time (in seconds) that the REST server waits for a valid HTTP request to reach a switch resource on a REST API connection (minimum 1; no maximum; default 30).
- `request-limit number` — Enter the number of concurrent requests allowed by the REST server (from 0 for disabled to any positive number for the number of concurrent requests supported; no maximum; default 0).
- `security-profile profile-name` — Enter the name of a security profile used by the REST API.

**i** **NOTE:** During an Enterprise SONiC upgrade or downgrade, locally configured users and their passwords and roles are properly migrated when installing a SONiC image using the `image install` command. The config migration scripts automatically migrate the `config_db.json`, `/etc/passwd`, `/etc/group`, `/etc/shadow`, `/etc/gshadow`, `/home/*`, and `/etc/sonic/cert/` directories, and `/var/spool/mail` files. However, if you reinstall Enterprise SONiC from ONIE, and manually migrate a configuration from one switch to another by copying and restoring a `config_db.json` file or by provisioning Enterprise SONiC using custom ZTP scripts, you must:

- Manually re-create or restore the certificate and private key files used for the REST and/or gNMI telemetry servers.
  - Manually reconfigure the local users using the `username password role` command or programmatic interfaces.
- Remotely authenticated users whose credentials are authenticated by RADIUS, TACACS+, or LDAP are not affected.

## Prerequisite: Install a host certificate

A host certificate is required when you use any of the REST API authentication modes — HTTP password, JWT token, and client certificate. Host certificates are used to securely identify a REST server and to establish encrypted connections between the REST service and clients that access the REST API.

By default, SONiC generates a self-signed local certificate to use with the REST server. To avoid interruptions in REST API operation, Dell Technologies recommends that you replace this auto-generated certificate with a host certificate that has been signed by a valid Certificate Authority (CA).

To use a CA-signed host certificate for REST API authentication:

1. (Optional) Create a host certificate request that you send to a Certificate Authority to receive a CA-signed certificate. Specify one of the following certificate-file and key-file locations:
  - `ftp://userid:passwd@hostip/filepath` — Installs a host certificate request on a remote FTP server.
  - `home://filename` — Installs a host certificate request in the home directory.
  - `http://hostip/filepath` — Installs a host certificate request on a remote HTTP server.
  - `scp://userid:passwd@hostip/filepath` — Installs a host certificate request on a remote SCP server.
  - `usb://filepath` — Installs a host certificate request on an attached USB device.

```
sonic# crypto cert generate request cert-file certificate-url key-file key-url
[password] [parameters]
```

You can add optional parameters to the host certificate request, such as:

- `altname` — A Subject Alternative Name, usually a DNS server name in the format: `DNS:server-name`.
- `cname` — A Common Name that identifies the certificate.

For detailed information on the optional parameters you can enter in a host certificate request, refer to the X.509 specification. An example of a host certificate request:

```
sonic# crypto cert generate request cert-file home://server-req.csr key-file home://
server.key cname myserver altname DNS:myserver
```

2. Install a host certificate-key pair on the switch from the specified URLs, where `certificate-url` and `key-url` are in one of these formats:
  - `ftp://userid:passwd@hostip/filepath` — Installs a host certificate file from a remote FTP server.
  - `home://filename` — Installs a host certificate file from the home directory.
  - `http://hostip/filepath` — Installs a host certificate file from a remote HTTP server.
  - `scp://userid:passwd@hostip/filepath` — Installs a host certificate file from a remote SCP server.

Enter an optional password if a private key file is password-protected. The certificate-key pair is maintained across image upgrades. Installing a host certificate triggers a certificate expiration check. To delete an installed certificate/key pair, use the `crypto cert delete` command.

```
sonic# crypto cert install cert-file certificate-url key-file key-url [password]
```

```
sonic# crypto cert install cert-file home://server.crt key-file home://server.key
Processing certificate ...
Installed host certificate
CommonName = server
IssuerName = www.dell.com
```

**(i) NOTE:** When you store a host certificate and key, the expiration date of the certificate is checked once a day. When the certificate expiration is within 30 days, a Syslog message is generated once a day. When the certificate expires in less than 14 days, Syslog warnings are generated. When the certificate expires, critical Syslog messages are generated. Some sample Syslog messages are:

```
Oct 05 21:39:46.356276+00:00 2021 sonic INFO system#monitor: Service mgmt-framework is using self-signed certificate /tmp/cert.pem consider using CA signed certificate instead.
Oct 05 21:39:46.356276+00:00 2021 sonic INFO system#monitor: Certificate /host_home/admin/cert.pem used by mgmt-framework is expiring in < 30 days.
Oct 05 21:39:46.356276+00:00 2021 sonic WARNING system#monitor: Certificate /host_home/admin/cert.pem used by telemetry is expiring in < 14 days.
Oct 05 21:39:46.356276+00:00 2021 sonic CRIT system#monitor: Certificate /host_home/admin/cert.pem used by mgmt-framework has expired.
Oct 05 21:39:46.356276+00:00 2021 sonic CRIT system#monitor: Failed to read certificate on mgmt-framework at /host_home/admin/cert.pem
Oct 05 21:39:46.356276+00:00 2021 sonic CRIT system#monitor: Certificate /host_home/admin/cert.pem used by mgmt-framework is not yet valid! Check your clock setting.
```

To check the status of an installed host certificate to see if it has expired, use the `crypto cert verify certificate-filename expiry` command; for example:

```
sonic# crypto cert verify server expiry
Certificate is valid!
```

3. Create a security profile for the REST service on the switch and associate the installed host certificate with the profile.

```
sonic# configure terminal
sonic(config)# crypto security-profile profile-name
sonic(config)# crypto security-profile certificate profile-name certificate-name
```

```
sonic(config)# crypto security-profile myserver
sonic(config)# crypto security-profile certificate myserver server
```

4. (Optional) Configure security profile settings.

- Require the REST API to verify if the key used to authenticate a remote device is associated with a CA-signed host certificate or a client certificate. Enter `True` to ensure that the correct certificate/key pair is used to access the switch; enter `False` not to check whether authentication is performed in host or client certificate mode. Default: `False`.

```
sonic(config)# crypto security-profile profile-name key-usage-check {True | False}
```

- Require the REST API service to verify if the remote device name matches the name on the certificate that is used to authenticate the device. `True` verifies the device name; `False` does not perform a remote device name check. Default: `False`.

```
sonic(config)# crypto security-profile profile-name peer-name-check {True | False}
```

- Require immediate revocation of an installed certificate if the revocation check returns a valid response. `True` performs a certificate check; `False` does not use certificate revocation. Default: `False`.

```
sonic(config)# crypto security-profile profile-name revocation-check {True | False}
```

- Add a global Certificate Revocation List (CRL) Distribution Point (CDP) list to receive CRL updates in addition to the CDPs defined in installed certificates. For *cdp-list*, enter a comma-separate list of the URLs for remote CDP servers in the format `http://host-ip/filepath`.

```
sonic(config)# crypto security-profile cdp-list profile-name cdp-list
```

For example:

```
sonic(config)# crypto security-profile cdp-list myserver http://a.example.com/cdp,http://b.example.com/cdp
```

- Add a global Online Certificate Status Protocol (OSCP) responder list in addition to the responders defined in installed certificates. For *oscp-list*, enter a comma-separate list of the URLs for remote OSCP responder servers in the format `http://host-ip/filepath`.

```
sonic(config)# crypto security-profile ocsp-list profile-name oscp-list
```

For example:

```
sonic(config)# crypto security-profile ocsp-list myserver http://a.example.com/ocsp,http://b.example.com/ocsp
```

5. Enable the security profile for the REST service. When the REST server restarts, it uses the new certificate.

```
sonic(config)# ip rest security-profile profile-name
```

```
sonic(config)# ip rest security-profile myserver
```

## Install a CA certificate for client certificate authentication

When the REST API uses client certificate authentication, it requires a client certificate to be sent by the client that accesses the switch. An installed CA certificate validates each client certificate. A client certificate must be signed by a certificate authority (CA) installed in the trust store and contain the common name (CN) field set to the name of the user.

**i | NOTE:** REST API servers that perform certificate authentication require that your remote device has a certificate and private key pair.

To use client certificate authentication, configure the REST server on the switch to accept password, JWT, and certificate authentication. Then install a CA certificate in the trust store and associate the trust store with a security profile used for client certificate verification:

1. Install a CA certificate on the switch from the specified URL, where *certificate-url* is one of these formats:

- `ftp://userid:passwd@hostip/filepath` — Installs a CA certificate file from a remote FTP server.
- `home://filename` — Installs a CA certificate file from the home directory.
- `http://hostip/filepath` — Installs a CA certificate file from a remote HTTP server.
- `scp://userid:passwd@hostip/filepath` — Installs a CA certificate file from a remote SCP server.

The CA certificate is maintained across image upgrades. Installing a CA certificate triggers a certificate expiration check. To delete an installed CA certificate, use the `crypto ca-cert delete` command.

```
sonic# crypto ca-cert install cert-file certificate-url
```

```
sonic# crypto ca-cert install cert-file home://GeoTrust_Universal_CA.crt
Processing certificate ...
Installed Root CA certificate
CommonName = GeoTrust Universal CA
IssuerName = GeoTrust Universal CA
```

**i** **NOTE:** An alternative way to install the CA certificate is to enter only the first part of the command `crypto ca-cert install` and then press Enter. When prompted, paste the raw ASCII format of the certificate between the BEGIN CERTIFICATE and END CERTIFICATE headers. For example:

```
sonic# crypto ca-cert install home://ca.crt
Processing certificate ...
Installed Root CA certificate as "ca"
CommonName = localhost
IssuerName = localhost
sonic# crypto cert delete all
sonic#
sonic# crypto
sonic# crypto ca-cert delete all
sonic# crypto ca-cert install
Certificate base file name: ca.crt
Paste certificate below.
Include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- headers.
Enter a blank line to abort this command.
Certificate:
-----BEGIN CERTIFICATE-----
MIIDRTCCAi2gAwIBAgIUFN1IQV72x5qbEgfVxi9t66SL54kwDQYJKoZIhvcNAQEL
BQAwFDESMBAGA1UEAwJBG9jYWxob3N0MB4XDTIZMDIyMTE3NTEzN1oXDTMzMDIx
ODE3NTEzN1owFDESMBAGA1UEAwJBG9jYWxob3N0MIIBIjANBgkqhkiG9w0BAQEFA
AOCAQ8AMIIIBCgKCAQEA8P6csTaX8FHPtMEgBxVncB2YcWLpJyuKxINlubVtjIjP
BhTHR4O2a01b380RInntrEI41Go5kOmaBB9eMs6XUIT+Gb1txRwV9j4cSYvNcz0
i89KogoN59q7325iliC2T/a+qs1XLtqPR5HvP1BfY8qX97vPZKv/Sd4iRaIrsqBq
tDgHkSPcUhevO/JG9jFhljA/vTAAnRZaTbS2JwzILkadqgIiWiTCUFI6K24NAkJuR
wcPEUWtJZbhoXIB2Y8jbBd0k+uASTCDr9ZB0leyC32GBtvGFoLWZpuuQNZ8DKlm
YuZjU4XwuEowlxFbkcIG+KtyNCrh6YvvvftUOFTYMaQIDAQABo4GOMIGLMAwGA1UD
EwQFMAMBAF8wHQYDVR0OBByEFI4b1JxDiooADE+XsYBDyspsNDMSME8GA1UDIWRI
MEAfI4b1JxDiooADE+XsYBDyspsNDMSoRikFjaUMRIwEAYDVQQDDAlsb2NhbgHv
c3SCFBTZSEFe9seamxIH1cYv+uki+eJMAsgAL1UdDwQEAWIBBjANBgkqhkiG9w0B
AQsFAAOCAQEAMhkFXtFmzN9sg8dISJ8afKNGTjVqpwkaVKMKyBaNmRB9Rn3qWp8V
i7r06vsQc+WmNc9PDQtazPRE1pJqBP1pb9kMCffNGwDF6p7GW6oTTtfPMLimrCL0
NVe31g8DiXiY5j2yT+0kdL6/h0+vM7VRjTVW9ODt+1IM/W5B2yTfxTU+j2Ok6oFc
eYs1bCF5aq6UOsKArqlsf600TcnA51FpqdA0ds89G7bpZg/nMXD2fHUEts362aZ
43tw01MUQ45EsONDWCtnWpunVI9jhzNqQZFhza30klHeTfGfmDdFjm1MRde8ljrc
mXsd4aQfR93FAE+RemvpOW9E/7Mw8D3GoQ==
-----END CERTIFICATE-----

Processing certificate ...
Installed Root CA certificate as "ca"
CommonName = localhost
IssuerName = localhost
```

To view an installed CA certificate, use the `show crypto ca-cert file` command. To verify the validity of an installed CA certificate, use the `crypto ca-cert verify expiry` command.

2. Create a trust store in which you store the CA certificates that are used to verify client certificates that access the REST API.

```
sonic# crypto trust-store trust-store-name
```

For example:

```
sonic(config)# crypto trust-store restts
```

3. Store a CA certificate in the trust store. Re-enter the command to add multiple CA certificates in the trust store.

```
sonic# crypto trust-store trust-store-name ca-cert certificate-name
```

For example:

```
sonic# configure terminal
sonic(config)# crypto trust-store restts ca-cert CA
```

4. Associate the trust store with a security profile used to authenticate REST API clients. To configure security profile settings, use the `crypto security-profile` command — see Steps 2 and 3 in the previous section "Install a host certificate for REST API authentication".

```
crypto security-profile trust-store profile-name trust-store-name
```

For example:

```
sonic(config)# crypto security-profile trust-store myserver restts
```

5. Enable the security profile for the REST service. When the REST server restarts, it uses the new certificate.

```
sonic(config)# ip rest security-profile profile-name
```

```
sonic(config)# ip rest security-profile myserver
```

This example shows how to install a client CA certificate, create a trust store with the CA certificate, and associate the CA certificate trust store with a security profile for REST server authentication:

```
sonic# crypto ca-cert install home://CA.crt
Processing certificate ...
Installed Root CA certificate
CommonName = www.dell.com
IssuerName = www.dell.com

sonic# configure terminal
sonic(config)# crypto trust-store myts ca-cert CA
sonic(config)# crypto security-profile trust-store myserver myts
```

From the remote device, access the REST API by specifying a certificate-key pair in a curl command:

```
curl -H "accept: application/yang-data+json" "https://switch-ip-address/restconf/data/openconfig-system:system/state" -k --key client.key --cert client.crt
```

A successful REST call with approved certificate authentication returns this response:

```
{"openconfig-system:state": {"boot-time": "1582791592", "current-datetime": "2020-02-28T02:59:29Z+00:00", "hostname": "st-sjc-z9264f-19"}}
```

**i | NOTE:** When you store a CA certificate, the expiration date of the certificate is checked once a day. When the certificate expiration is within 30 days, a Syslog message is generated once a day. When the certificate expires in less than 14 days, Syslog warnings are generated. When the certificate expires, critical Syslog messages are generated.

To check the status of a CA certificate to see if it has expired, use the `crypto cert verify certificate-filename expiry` command; for example:

```
sonic# crypto cert verify CA.crt expiry
Certificate is valid!
```

## Use HTTP password authentication

A curl command encodes the user credentials in the `--user` option with a `username:password` value:

```
curl --user admin:sonicadmin -k https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces/interface=Eth1%2F2
```

## Use JWT token authentication

A JWT token is valid for 1 hour, with a refresh interval of 30 seconds. You can only refresh the token at most 30 seconds before it expires. If the token expires, you must reauthenticate your REST API session.

To generate a JWT token, send the following curl command to the switch from a remote device:

```
curl -k -X POST https://switch-ip-address/authenticate -d '{"username": "admin", "password": "sonicadmin"}'
```

The switch returns a response that contains the JWT access code to use instead of username and password to authenticate REST API calls on the switch:

```
{"access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmcFtZSI6ImFkbWluIiwicm9sZXMiOlsiYWRtaW4iLCJzdWRvIiwiZG9ja2VyIl0sImV4cCI6MTU4MjI0NTA0M30.3wRyN5FfN3LIg2hTUErm3qT5NQEoCNPxQxrRz3PcWDg", "token_type": "Bearer", "expires_in": 3600}
```

To use the new JWT token to access the REST API and retrieve data about switch resources; for example, openconfig-interfaces:

```
curl -k -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmcFtZSI6ImFkbWluIiwicm9sZXMiOlsiYWRtaW4iLCJzdWRvIiwiZG9ja2VyIl0sImV4cCI6MTU4MjI0NTA0M30.3wRyN5FfN3LIg2hTUErm3qT5NQEoCNPxQxrRz3PcWDg" https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces/interface=Eth1%2F2
```

You can refresh a JWT token only valid during the refresh interval before the expiration time ends. To refresh the token, copy the current access code into the refresh curl command syntax:

```
curl -k -X POST https://switch-ip-address/refresh -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmcFtZSI6ImFkbWluIiwicm9sZXMiOlsiYWRtaW4iLCJzdWRvIiwiZG9ja2VyIl0sImV4cCI6MTU4MjI0NTA0M30.3wRyN5FfN3LIg2hTUErm3qT5NQEoCNPxQxrRz3PcWDg"
```

## View REST API authentication

To view the authentication modes configured for REST API authentication:

```
sonic# show ip rest authentication

REST Client Authentication Modes

client_auth: password,jwt
```

To display the settings used in REST API authentication:

```
sonic# show ip rest
Log level is not-set
JWT valid is 3600 seconds
JWT refresh is 900 seconds
Port is 443
Request limit is 0
Read timeout is 30 seconds
Client authentication mode is password
Security profile is not-set
```

To display information about an installed host certificate:

```
sonic# show crypto cert {certificate-name | all}
```

```
sonic# show crypto cert server
Certificate Name: server
Certificate:
 Data:
 Version: 3 (0x2)
 Serial Number: 2 (0x2)
 Signature Algorithm: sha256WithRSAEncryption
 Issuer: C = US, ST = CA, L = Sacramento, O = Dell, OU = Networking, CN =
www.dell.com
 Validity
 Not Before: Aug 13 08:00:00 2015 GMT
```

```

 Not After : Aug 13 09:00:00 2025 GMT
Subject: C = US, ST = CA, O = Dell, OU = Networking, CN = server
Subject Public Key Info:
 Public Key Algorithm: rsaEncryption
 RSA Public-Key: (2048 bit)
 Modulus:
 00:c1:cd:48:73:ad:62:5d:fb:70:f0:7d:eb:20:97:
 e5:5a:5c:a7:89:ef:0e:12:c4:36:52:1c:fa:d1:a1:
 91:1d:f3:95:2a:f8:c6:ed:41:da:bf:e8:a2:d2:14:
 f8:6d:7f:3d:c0:ae:58:08:91:75:82:8e:3e:3e:6f:
 4c:04:e2:86:75:45:a0:e8:37:5f:b2:92:81:b0:23:
 34:eb:15:c4:d6:69:f1:c6:93:9e:a7:32:b9:52:f8:
 c1:53:35:57:ec:70:fb:85:dd:24:8c:47:6b:2d:34:
 9c:03:60:ad:a6:de:f3:88:1b:17:16:97:b0:e0:09:
 17:67:ed:4d:c3:a6:41:70:e9:86:be:f1:2c:b5:14:
 0a:c3:45:58:96:7f:73:43:30:35:3a:7a:42:8c:53:
 df:bb:de:fe:58:50:2b:83:df:71:65:41:ff:ae:30:
 e7:ce:f6:99:15:5f:ad:d5:e8:86:e0:18:80:a0:d0:
 e9:01:45:7b:4e:51:7d:38:bb:e3:25:9c:5c:9c:b7:
 20:ea:ff:4e:aa:65:e2:51:4a:c3:4b:82:b8:4f:85:
--more--
 e4:af:e1:b6:5c:7f:7e:90:a9:29:1a:b9:e0:5b:d6:
 b1:cd:2f:7a:89:38:ad:6f:97:66:de:cf:89:b1:c8:
 46:05:5d:47:7d:33:a2:c9:77:22:a4:65:82:9a:2d:
 c5:ed
 Exponent: 65537 (0x10001)
X509v3 extensions:
 Authority Information Access:
--more--
 OCSP - URI:http://127.0.0.1:8181
X509v3 Subject Alternative Name:
 DNS:server
Signature Algorithm: sha256WithRSAEncryption
 66:84:b0:d5:18:32:2b:7b:22:c4:8f:e4:b1:c1:c8:bd:ce:ca:
 33:5b:4b:4a:79:ea:d3:9c:8b:20:d9:46:2f:7e:6f:91:1c:89:
 a2:2e:14:bc:25:71:1a:05:a6:1a:7d:f9:3a:ef:93:55:29:bc:
 4a:87:f3:96:96:93:27:c3:7a:43:9c:f8:a2:1a:c4:0a:e5:95:
 a7:00:b8:04:1c:70:25:c5:3c:9c:ed:da:8e:fd:d3:b1:8e:76:
 97:84:df:59:c3:8e:11:22:10:23:97:71:ff:14:31:1c:b8:a0:
--more--
 ed:1f:e1:31:c7:c1:de:89:f6:3e:0b:d5:95:84:ee:4b:64:ed:
 ab:0a:f7:36:44:12:52:d1:36:10:96:7b:ef:4c:44:b6:46:b2:
--more--
 41:20:0b:f5:70:b9:68:f1:34:5a:78:71:73:77:d2:8b:f9:5d:
 2f:ea:dd:14:b0:71:0f:9d:73:e0:72:06:b7:a2:0c:60:87:83:
--more--
 91:8a:4a:2d:cc:7a:3b:40:22:b9:3d:7a:d8:26:03:7e:2b:a3:
 8a:59:34:07:30:90:9c:17:55:11:ce:f7:b4:6b:44:2d:f2:bb:
 9a:e4:7c:fe:09:fb:db:22:d1:1f:b2:e7:7a:e4:e1:98:a4:d2:
 ef:27:a3:37:30:94:5c:09:c2:71:5e:bf:5b:ac:a9:af:ee:ae:
--more--
 ac:63:83:d3

```

To display information about an installed CA certificate:

```

sonic# show crypto ca-cert {certificate-name | all}

sonic# show crypto ca-cert CA
Certificate Name: CA
Certificate:
 Data:
 Version: 3 (0x2)
 Serial Number:
 64:f2:8f:46:09:68:73:34:46:d4:53:94:2c:fc:1f:12:06:6c:ee:8c
 Signature Algorithm: sha256WithRSAEncryption
 Issuer: C = US, ST = CA, L = Sacramento, O = Dell, OU = Networking, CN =
www.dell.com
 Validity
 Not Before: Jul 7 20:24:34 2022 GMT
 Not After : Jul 4 20:24:34 2032 GMT
 Subject: C = US, ST = CA, L = Sacramento, O = Dell, OU = Networking, CN =
www.dell.com

```

```

Subject Public Key Info:
 Public Key Algorithm: rsaEncryption
 RSA Public-Key: (2048 bit)
 Modulus:
 00:ea:27:4f:1e:9f:03:97:3f:77:f1:48:6a:29:46:
 86:ef:d9:d1:ec:e1:d0:93:28:88:f6:36:47:66:21:
 00:92:d9:d3:65:8d:3c:bc:80:ac:8f:22:ee:cd:20:
 5e:ec:47:13:e1:a8:85:20:01:73:30:94:8c:6a:0a:
 73:93:3c:0a:e5:f6:b1:0a:f5:ad:c1:0b:e2:84:3c:
 a6:5b:5f:7d:b8:71:da:ab:46:88:03:e8:42:63:52:
 76:8f:1d:3e:1a:c1:d4:dc:55:75:b8:8f:af:03:19:
 95:d3:84:62:e9:f9:1b:6c:9d:ff:a7:f6:8e:32:2f:
 32:7f:5d:05:79:84:e3:f4:cd:78:59:49:80:18:a8:
 d6:c5:6d:54:2d:07:1a:16:e0:ce:15:f4:c6:2c:ae:
 f5:8f:a1:89:aa:37:79:ea:1a:75:dd:11:df:ee:8a:
 87:ed:4e:3c:3a:54:96:d4:13:75:a0:af:57:94:b0:
 d5:13:80:3d:de:54:9b:e2:56:6b:1e:a1:d2:fd:93:
 d2:3b:77:7c:b7:d8:e9:7d:d0:5d:a1:dc:89:0e:a6:
 c4:82:e6:fc:15:4c:6a:df:45:8c:ab:0b:07:cf:49:
 b7:53:ee:2a:84:5a:ac:29:09:7c:ee:5a:ff:94:19:
 e2:7b:3f:11:7b:f7:dd:5b:2a:da:99:98:59:b8:07:
 23:e9
 Exponent: 65537 (0x10001)
X509v3 extensions:
 X509v3 Basic Constraints:
 CA:TRUE
 X509v3 Key Usage:
 Digital Signature, Non Repudiation, Key Encipherment, Certificate Sign
Signature Algorithm: sha256WithRSAEncryption
 63:9a:e3:b4:b7:10:d7:fa:92:cd:04:7b:f4:6a:c3:5e:7c:11:
 c6:e6:9e:eb:61:60:97:cf:ad:60:03:ab:89:0f:e2:f1:5b:94:
 7f:dd:6e:21:e9:21:60:0a:8d:81:ec:7e:35:84:b6:97:5e:65:
 07:c4:e2:44:7c:8e:8c:7e:20:94:29:6d:32:e7:dc:6c:70:5d:
 64:ed:cc:3a:d8:83:11:84:26:ab:87:f2:ce:05:8f:29:d2:0f:
 62:de:a2:d9:86:cb:d4:7f:1c:60:f8:38:e3:41:9b:13:6d:24:
 b4:eb:0d:ac:8b:5f:96:d4:0a:5c:26:aa:20:d3:c0:f8:06:24:
 84:3d:d0:1b:e0:33:99:49:5e:3c:f9:77:68:94:d7:f2:11:be:
 39:41:2a:44:5d:9e:a6:b6:a7:02:14:f4:02:6f:f1:f8:71:a5:
 c2:ac:94:39:5d:b6:68:7d:00:5a:e5:92:74:5c:f7:52:5e:2d:
 6a:4f:a6:0c:a6:1b:c1:ff:a9:46:1f:3c:5e:a1:16:fa:72:55:
 1b:84:d2:a8:25:b1:c8:f2:35:97:e0:02:2c:08:9c:e3:69:0e:
 2e:0d:9c:f1:98:25:28:06:dc:57:59:9d:bb:48:97:02:63:16:
 80:80:b9:1e:5d:13:10:a7:8a:1c:84:2d:aa:7d:ec:3e:67:a0:
 14:b5:d9:6a

```

To display the security profiles used in REST API authentication:

```

sonic# show crypto security-profile

Security Profile : myserver

Key Usage Check : False
Peer Name Check: False
Revocation Check: False
Certificate Name: None
Trust Store: None
CDP Identifiers: http://a.example.com/cdp,http://b.example.com/cdp
OCSP Responders: http://a.example.com/ocsp,http://b.example.com/ocsp

Security Profile : mysp

Key Usage Check : False
Peer Name Check: False
Revocation Check: False
Certificate Name: server
Trust Store: None
CDP Identifiers: None
OCSP Responders: None

```

To display the trust stores used for REST API authentication:

```
sonic# show crypto trust-store

Trust Store : myts

CA Certificates Name: CA
```

## REST API requests using curl

Using the REST API, you can provision switches using HTTPS requests. The examples in this section show how to access the REST API using curl commands. Curl is a Linux shell command that generates HTTPS requests and is run on an external server.



### Curl commands

- **-X** — specifies the HTTPS request type; for example, `POST`, `PATCH`, or `GET`.
- **-u** — specifies the username and password to use for server authentication.
- **-k** — runs a REST request in insecure mode. Insecure mode does not verify a server certificate. Although an encrypted HTTPS connection is used, the server certificate is self-signed and not verified.
- **-H** — specifies an extra header to include in the request when sending HTTPS to a server. You can enter multiple extra headers.
- **-d** — sends the specified data in an HTTPS request.

Use `%2F` to represent a forward slash (/); for example, enter the IP address and prefix length `10.1.2.3/24` as `10.1.2.3%2F24`.

For more information, see the [curl Manpage](#).

### REST API operations

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>POST</b>   | Creates data only if the data does not exist.                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>PUT</b>    | Replaces data if it exists, or creates data if the data does not exist.                                                                                                                                                                                                                                                                                                                                                           |
| <b>PATCH</b>  | Creates or updates data, but does not delete it.                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>DELETE</b> | Deletes data if it exists.                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>GET</b>    | Retrieves configuration and operational data. The following parameters are supported in a GET request: <ul style="list-style-type: none"><li>• <b>content</b> — Specifies the types of child resources to retrieve: configuration and/or non-configuration.</li><li>• <b>depth</b> — Limits the number of levels of child resources that are returned.</li><li>• <b>fields</b> — Specifies the data fields to retrieve.</li></ul> |

### Usage information

Take into account the following items when you access the REST API using curl commands:

- In an HTTPS request, it is recommended that you use the most specific URI that is available to avoid unnecessary data and delay in the response.
- When a REST query is in progress, you cannot configure any CLI commands until a REST query is complete.

# YANG PATCH operation

The REST API supports YANG PATCH requests as described in RFC 8072. In a YANG PATCH operation, multiple resources are edited using the same HTTPS request. The RESTCONF server supports create, delete, merge, replace, and remove edit operations. Multiple edit operations are processed in the user-defined order.

## YANG PATCH restrictions

- In Enterprise SONiC, a YANG PATCH request does not support move and insert edit operations.
- A YANG PATCH request cannot run on a datastore resource; that is, a target resource cannot be at the restconf/data subtree level.

## YANG PATCH example

To use a YANG PATCH operation to apply edits on the switch to create an ACL with one rule:

1. Create an ACL entry with `edit1`.
2. Create a rule in the ACL entry with `edit2`.
3. Send the PATCH HTTP request from a RESTCONF client.

For example:

```
PATCH /restconf/data/openconfig-acl:acl/acl-sets \
 HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-patch+json
{
 "ietf-yang-patch:yang-patch": {
 "patch-id": "ACL Rules",
 "edit": [
 {
 "edit-id": "edit1",
 "operation": "create",
 "target": "/acl-set=MyACL1,ACL_IPV4",
 "value": {
 "acl-set": [
 {
 "name": "MyACL1",
 "type": "ACL_IPV4",
 "config": {
 "name": "MyACL1",
 "type": "ACL_IPV4",
 "description": "Description for MyACL1"
 }
 }
]
 }
 },
 {
 "edit-id": "edit2",
 "operation": "create",
 "target": "/acl-set=MyACL1,ACL_IPV4/acl-entries/acl-entry=1",
 "value": {
 "acl-entry": [
 {
 "sequence-id": 1,
 "config": {
 "sequence-id": 1,
 "description": "Description for Changed Rule"
 },
 "ipv4": {
 "config": {
 "source-address": "13.1.1.1/32",
 "destination-address": "23.1.1.1/32",
 "protocol": "IP_TCP"
 }
 },
 "transport": {
 "config": {
 "source-port": 102,
 "destination-port": 102
 }
 }
 }
]
 }
 }
]
 }
}
```

```
 }
 },
 "actions": {
 "config": {
 "forwarding-action": "ACCEPT"
 }
 }
}
]
```

The response from the RESTCONF server:

```
HTTP/1.1 200 OK
Date: Thu, 26 Jan 2017 20:56:30 GMT
Content-Type: application/yang-data+json
{
 "ietf-yang-patch:yang-patch-status": {
 "patch-id": "ACL Rules",
 "ok": [null]
 }
}
```

## REST API examples

Some common REST API operations include retrieving interface information and configuring interface settings, such as MTU. These examples use curl commands to send the HTTPS request. It is recommended that you use the Swagger user interface to show all REST endpoints and then test them in a web browser.

**(i) NOTE:** In each REST API request, use the IP address of the Management interface. The following examples use the Management IP address 10.11.56.39.

### Get information about all interfaces

```
curl -k -u "admin:sonicadmin" -X GET "https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces" -H "accept: application/yang-data+json"
```

### Get information about a specified interface

```
curl -k -u "admin:sonicadmin" -X GET "https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces/interface=Eth1%2F3" -H "accept: application/yang-data+json"
```

### Configure interface MTU

```
curl -k -u "admin:sonicadmin" -X PATCH "https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces/interface=Eth1%2F1/config/mtu" -H "accept: application/yang-data+json" -H "Content-Type: application/yang-data+json" -d "{ \"openconfig-interfaces:mtu\": 9105 }"
```

### Create an ACL

```
curl -k -u "admin:sonicadmin" -X POST "https://switch-ip-address/restconf/data/openconfig-acl:acl/acl-sets" -H "accept: application/yang-data+json" -H "Content-Type: application/yang-data+json" -d "{\"openconfig-acl:acl-set\": [{\"name\": \"MyACL1\",
```

```
{"type": "ACL_IPV4", "config": {"name": "MyACL1", "type": "ACL_IPV4", "description": "Description for MyACL1"}}]}
```

## Get ACL information

A POST request does not return a response. You must send a GET request to verify the configuration change; for example, to receive information for all ACLs:

```
curl -k -u "admin:sonicadmin" -X GET "https://switch-ip-address/restconf/data/openconfig-acl:acl-sets" -H "accept: application/yang-data+json"
{
 "openconfig-acl:acl-sets": {
 "acl-set": [
 {
 "config": {
 "description": "Description for MyACL1",
 "name": "MyACL1",
 "type": "openconfig-acl:ACL_IPV4"
 },
 "name": "MyACL1",
 "state": {
 "description": "Description for MyACL1",
 "name": "MyACL1",
 "type": "openconfig-acl:ACL_IPV4"
 },
 "type": "openconfig-acl:ACL_IPV4"
 }
]
 }
}
```

To receive information about a specific ACL:

```
curl -k -u "admin:sonicadmin" -X GET "https://switch-ip-address/restconf/data/openconfig-acl:acl-sets/acl-set=MyACL1,ACL_IPV4" -H "accept: application/yang-data+json"
{
 "openconfig-acl:acl-set": [
 {
 "config": {
 "description": "Description for MyACL1",
 "name": "MyACL1",
 "type": "openconfig-acl:ACL_IPV4"
 },
 "name": "MyACL1",
 "state": {
 "description": "Description for MyACL1",
 "name": "MyACL1",
 "type": "openconfig-acl:ACL_IPV4"
 },
 "type": "openconfig-acl:ACL_IPV4"
 }
]
}
```

## Create VLAN10

```
curl -k -u "admin:sonicadmin" -X POST "https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces" -H "accept: application/yang-data+json" -H "Content-Type: application/yang-data+json" -d "{ \"openconfig-interfaces:interface\" : [{ \"config\" : { \"name\" : \"Vlan10\" }, \"name\" : \"Vlan10\" }] }
```

## Delete VLAN10

```
curl -k -u "admin:sonicadmin" -X DELETE "https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces/interface=Vlan10" -H "accept: application/yang-data+json"
```

## Add Eth1/10 interface to VLANs 2, 4, and 5

```
curl -k -u "admin:sonicadmin" -X PATCH "https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces/interface=Eth1%2F10/openconfig-if-ethernet:ethernet/openconfig-vlan:switched-vlan/config/trunk-vlans" -H "accept: application/yang-data+json" -H "Content-Type: application/yang-data+json" -d "{ \"openconfig-vlan:trunk-vlans\": [2,4,5] }"
```

## Remove Eth1/10 interface from VLAN4

```
curl -k -u "admin:sonicadmin" -X DELETE "https://switch-ip-address/restconf/data/openconfig-interfaces:interfaces/interface=Eth1%2F10/openconfig-if-ethernet:ethernet/openconfig-vlan:switched-vlan/config/trunk-vlans=4" -H "accept: application/yang-data+json"
```

## Configure BGP sessions only on IPv6 links

Using the REST API, you can configure `v6only` for a neighbor so that BGP sessions are established only on the neighbor's IPv6 link local address; for example:

```
curl -X PATCH https://switch-ip-address/restconf/data/openconfig-network-instance:network-instances/network-instance={name}/protocols/protocol=BGP,bgp/bgp/neighbors/neighbor={interface_name}/config/openconfig-bgp-ext:v6only -H accept: */* -k -u admin:[Password] -H Content-Type: application/yang-data+json -d {"openconfig-bgp-ext:v6only": true }
```

## Get power supply status

```
curl -kX GET "https://switch-ip-address/restconf/data/openconfig-platform:components/component=PSU%201/power-supply/state" -H "accept: application/yang-data+json" -H "authorization: Basic YWRtaW46c29uaWMxMjM="
```

The GET response contains power supply output voltage in encoded form; for example:

```
REST GET Command:
curl -kX GET "https://switch-ip-address/restconf/data/openconfig-platform:components/component=PSU%201/power-supply/state/openconfig-platform-psu:output-voltage" -H "accept: yang-data+json" -H "authorization: Basic YWRtaW46c29uaWMxMjM="

REST GET response:
{ "openconfig-platform-psu:output-voltage": "QUMKPQ==" }
```

The **QUMKPQ==** value of the PSU output voltage is base64-encoded. To decode the response, use base64 decode, and copy the decoded 4 bytes into a buffer in Big-Endian (Network) order. The following Python example shows how to decode the voltage value 12.1899995803833:

```
admin@sonic:~$ python
Python 2.7.13 (default, Sep 26 2018, 18:42:22)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import base64
```

```
>>> import struct
>>> struct.unpack('>f', base64.b64decode(b'QUMKPQ=='))
(12.1899995803833,)
```

 **NOTE:** To view power supply status from the command-line interface, use the show platform psusummary command.

## Clear counters on Eth1/1 interface

The following example is a Remote Procedure Call (RPC) operation from a remote device to clear the interface counters for the Eth1/1 interface. All RPC operations use the POST action and require that input data is in JSON format.

```
$ curl -k -X POST https://localhost/restconf/operations/openconfig-interfaces-ext:clear-counters -u admin:sonicadmin -d '{"openconfig-interfaces-ext:input": {"interface-param": "Eth1%2F1"}}' -H "Content-Type: application/yang-data+json"
```

The output that is returned is:

```
{
 "openconfig-interfaces-ext:output": {
 "status": 0,
 "status-detail": "Success: Cleared Counters"
 }
}
```

## Get system firmware

To retrieve information about a firmware component in a REST API call:

1. Enter the show platform firmware command and note the number in which the component displays in alphabetical order. See [Show system firmware](#) for a description of how to identify the number of a component in the show output.
2. In the GET curl command, enter the number of the component in the component=FIRMWARE%20# parameter, where # is a number from 1 to 6. In the following example, component=FIRMWARE%201 identifies BIOS as the first component listed in show platform firmware output for the switch.

```
$ curl -k -u admin:sonic123 -X GET "https://switch-ip-address/restconf/data/openconfig-platform:components/component=FIRMWARE%201" -H "accept: application/yang-data+json" |
python -m json.tool
{
 "openconfig-platform:component": [
 {
 "chassis": {
 "state": {
 "openconfig-platform-ext:name": "Z9264F-ON"
 }
 },
 "config": {
 "name": "FIRMWARE 1"
 },
 "name": "FIRMWARE 1",
 "state": {
 "description": "Performs initialization of hardware components during booting",
 "firmware-version": "3.23.0.0-6",
 "name": "BIOS"
 }
 }
]
}
```

## GET query parameters

### content

Use the `content` parameter to select the types of data resources to retrieve. The possible values are `config`, `nonconfig`, or `all`. The default is `all`.

- A GET `content=all` request returns both configuration (ReadWrite) and non-configuration (ReadOnly) data resources.

```
curl -X GET "https://localhost/restconf/data/openconfig-system:system/dns?content=all" -H "accept: application/yang-data+json" --insecure
```

For example, the returned output for a GET `content=all` request is:

```
{
 "openconfig-system:dns": {
 "config": {
 "openconfig-system-ext:source-interface": "Ethernet0"
 },
 "state": {
 "openconfig-system-ext:source-interface": "Ethernet0"
 },
 "servers": {
 "server": [
 {
 "address": "8.8.8.8",
 "config": {
 "address": "8.8.8.8"
 },
 "state": {
 "address": "8.8.8.8"
 }
 },
 {
 "address": "8::8",
 "config": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 },
 "state": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 }
 }
]
 }
 }
}
```

- A GET `content=config` request returns only `config` (ReadWrite) data resources.

```
curl -X GET "https://localhost/restconf/data/openconfig-system:system/dns?content=config" -H "accept: application/yang-data+json" --insecure
```

```
{
 "openconfig-system:dns": {
 "config": {
 "openconfig-system-ext:source-interface": "Ethernet0"
 },
 "servers": {
 "server": [
 {
 "address": "8.8.8.8",
 "config": {
 "address": "8.8.8.8"
 }
 },
 {
 "address": "8::8",
 "config": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 }
 }
]
 }
 }
}
```

```
 }
 }
}
```

- A GET content=nonconfig request returns only nonconfig (ReadOnly) data resources.

```
curl -X GET "https://localhost/restconf/data/openconfig-system:system/dns?content=nonconfig" -H "accept: application/yang-data+json" --insecure
```

```
{
 "openconfig-system:dns": {
 "state": {
 "openconfig-system-ext:source-interface": "Ethernet0"
 },
 "servers": {
 "server": [
 {
 "address": "8.8.8.8",
 "state": {
 "address": "8.8.8.8"
 }
 },
 {
 "address": "8::8",
 "state": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 }
 }
]
 }
 }
}
```

## depth

Use the depth parameter to limit the number of levels returned in the Get request. The possible values are unbounded or 1...65535. The default is unbounded.

- A GET depth=unbounded request returns all data resources.

```
curl -X GET "https://localhost/restconf/data/openconfig-system:system/dns?depth=unbounded" -H "accept: */*" -H "Content-Type: application/yang-data+json" --insecure
```

For example, the returned output for a GET depth=unbounded request is:

```
{
 "openconfig-system:dns": {
 "config": {
 "openconfig-system-ext:source-interface": "Ethernet0"
 },
 "state": {
 "openconfig-system-ext:source-interface": "Ethernet0"
 },
 "servers": {
 "server": [
 {
 "address": "8.8.8.8",
 "config": {
 "address": "8.8.8.8"
 },
 "state": {
 "address": "8.8.8.8"
 }
 },
 {
 "address": "8::8",
 "config": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 }
 }
]
 }
 }
}
```

```

 },
 "state": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 }
 }
}
}
}
}
}

```

- A GET depth=4 request returns the target resource and three data resource layers.

```
curl -X GET "https://localhost/restconf/data/openconfig-system:system/dns?depth=4" -H "accept: */*" -H "Content-Type: application/yang-data+json" --insecure
```

```
{
 "openconfig-system:dns": {
 "config": {
 "openconfig-system-ext:source-interface": "Ethernet0"
 },
 "state": {
 "openconfig-system-ext:source-interface": "Ethernet0"
 },
 "servers": {
 "server": [
 {
 "address": "8.8.8.8"
 },
 {
 "address": "8::8"
 }
]
 }
 }
}
```

## fields

Use the `fields` parameter to specify the data resource to retrieve.

For example, note the following GET request without a `fields` parameter and the data returned:

```
curl -X GET "https://localhost/restconf/data/openconfig-system:system/dns/servers/server" -H "accept: application/yang-data+json" --insecure
```

Returned output:

```
{
 "openconfig-system:server": [
 {
 "address": "8.8.8.8",
 "config": {
 "address": "8.8.8.8"
 },
 "state": {
 "address": "8.8.8.8"
 }
 },
 {
 "address": "8::8",
 "config": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 },
 "state": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 }
 }
]
}
```

- To use the `fields` parameter to retrieve the leaf inside the `config` container:

```
curl -X GET "https://localhost/restconf/data/openconfig-system:system/dns/servers/server=8::8/config?fields=openconfig-system-ext:vrf-name" -H "accept: application/yang-data+json" --insecure
```

This curl command returns the following data:

```
{
 "openconfig-system:config": {
 "openconfig-system-ext:vrf-name": "mgmt"
 }
}
```

- To use the `fields` parameter to retrieve the leaf inside the `config` container and the contents of the `state` container:

```
curl -X GET "https://localhost/restconf/data/openconfig-system:system/dns/servers/server?fields=config/openconfig-system-ext:vrf-name, state" -H "accept: application/yang-data+json" --insecure
```

This curl command returns the following data:

```
{
 "openconfig-system:server": [
 {
 "address": "8.8.8.8",
 "state": {
 "address": "8.8.8.8"
 }
 },
 {
 "address": "8::8",
 "config": {
 "openconfig-system-ext:vrf-name": "mgmt"
 },
 "state": {
 "address": "8::8",
 "openconfig-system-ext:vrf-name": "mgmt"
 }
 }
]
}
```

# gRPC Network Management Interface

**i** **NOTE:** The gRPC Network Management Interface (gNMI) is supported in the Cloud Standard, Cloud Premium, Enterprise Standard, and Enterprise Premium bundles. gNMI is supported in the Edge Standard bundle, but only on E-Series switches.

The gRPC Network Management Interface allows you to configure and monitor switches using remote procedure calls (RPCs). gNMI supports both read/write configuration, and telemetry streaming of configuration and operational data using gNOI RPCs. gNMI uses JavaScript object notation (JSON) to code data for YANG data objects and complies with RFC 7951.

To create RPCs, you can use gNMI RPC code in your own client scripts and software written in various programming languages, such as C, C++, C#, Java, JavaScript, PERL, and Python. Use the gNMI/gRPC libraries and the protobuf files that are available in the Enterprise SONiC OS data model.

In addition, for a complete list of Enterprise SONiC YANG models, go to the [Dell Technologies Support](#) site, search for Enterprise SONiC Distribution, open the Drivers & Downloads page, select a software release, and download the **Enterprise SONiC OS datamodel** zip file.

## gNMI requests

gNMI supports four types of requests:

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Get</b>          | Retrieves configuration and operational data from YANG objects on the switch. A GetRequest is sent from a remote device to the target switch. The switch sends a GetResponse in response to the request. In the <code>type</code> field of a GetRequest, the following parameters are supported. Only one parameter is supported at a time in the <code>type</code> field of a GetRequest. <ul style="list-style-type: none"> <li>• <b>CONFIG</b> — Specifies read/write data on the target switch. If the data schema is described in YANG, the <code>Get type=CONFIG</code> corresponds to the <code>config true</code> set of leaves on the target.</li> <li>• <b>STATE</b> — Specifies read-only data on the target switch. If the data schema is described in YANG, the <code>Get type=STATE</code> corresponds to the <code>config false</code> set of leaves on the target.</li> <li>• <b>OPERATIONAL</b> — Specifies read-only data on the target switch that is related to software processes operating on the device, or external interactions of the device.</li> </ul> |
| <b>Set</b>          | Updates, replaces, or deletes configuration values in the data tree of a YANG model. An update request modifies the configuration parameters with the values that are specified in the JSON request. A replace request resets the configuration parameters to their default settings. A delete request removes the configured value and resets it to the default. The update and replace requests create an object if it does not exist.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Capabilities</b> | Returns the gNMI version and a list of supported YANG models and encodings.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Subscribe</b>    | Allows a client to subscribe for updates to one or more paths.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

## gNMI authentication

There are three types of authentication which you can include in gNMI requests:

- Username and password — The username and password are sent in the metadata in the request,
- JSON web token (JWT) — JWT requires you to first authenticate using a gNOI RPC call by providing a username and password. The RPC returns a token for the specified user that is valid for the configured expiry time. You must refresh the token within the configured refresh period before expiration, otherwise you must reauthenticate.
- Certificates — Certificate authentication requires the use of a valid certificate that is signed by the certificate authority (CA) specified in the switch, and must contain the username in the common name (CN) field. For more information, see [gNMI certificate authentication](#).

By default, HTTP password and JWT are enabled for gNMI authentication on a switch. To verify the currently enabled gNMI authentication modes:

```
sonic# show ip telemetry authentication
```

```
Telemetry Client Authentication Modes

client_auth: password,jwt
```

To reconfigure the gNMI authentication modes:

```
sonic(config)# ip telemetry authentication auth-mode
```

Where *auth-mode* is one or more of the following values that are separated by commas:

- *password* — Enable HTTP password authentication.
- *jwt* — Enable JWT token-based authentication.
- *cert* — Enable certificate-based authentication.
- *none* — Disable all authentication modes.

Enter multiple values for *auth-mode* by separating them with a comma; for example:

```
sonic(config)# ip telemetry authentication password,jwt,cert
```

**(i) NOTE:** Username and certificate authentication require a reauthentication of the user and role for each request. To avoid this additional overhead, use JWT authentication. JWT-based authentication uses a token to authenticate the user and role one time for the first request in the entire session.

To reconfigure the settings used in gNMI authentication and other Telemetry services, use the `ip telemetry` command.

```
sonic(config)# ip telemetry {jwt-refresh seconds | jwt-valid seconds | log-level severity-level | port number | security-profile profile-name}
```

- *jwt-refresh seconds* — Enter the time (in seconds) before a JWT token can be refreshed (minimum 0; no maximum; default 3600).
- *jwt-valid seconds* — Enter the time (in seconds) that a JWT token is valid (minimum 0; no maximum; default 3600).
- *log-level severity-level* — Enter the severity level of messages to be logged for debugging (1-7), where lower numbers indicate more severe conditions: alerts for immediate action (1), critical conditions (2), errors (3), warnings (4), notifications (5), informational (6), debugging (7); no default.
- *port number* — Enter the TCP port number used by the telemetry server to receive gNMI requests (1-65535; default 8080).
- *security-profile profile-name* — Enter the name of a security profile used by the REST API.

### JSON encoding

RFC 7951 specifies the rules for using JSON to encode YANG data values, such as leafs, containers, and nodes. If you manually build the JSON payload, use the REST swagger user interface to obtain a JSON template and enter the template. You can also follow these methods:

- Fill out the `protobuf` object using a supported programming language, and then serialize it to JSON.
- Send a gNMI Get request on the path you want to set, then modify the gNMI Get response and send the modified output in a Set request.

**(i) NOTE:** A gNMI Set request may not return output, such as when you create an object as an IP address. If you manually construct the JSON payload, use a tool like PYANG to print the YANG model tree. Use the data tree to build the JSON content. For example, to add a new IPv4 address, the JSON coding is:

**gNMI Set request** `openconfig-interfaces:interfaces/interface[name=Eth1/9]/subinterfaces/path`

```
{
 "openconfig-if-ip:ipv4": {
 "addresses": {
 "address": [
 {
 "ip": "9.9.9.9",
 "config": {
 "ip": "9.9.9.9",
 "prefix-length": 24
 }
 }
]
 }
 }
}
```

### JSON content

```
]
 }
}
```

## Topics:

- gNMI certificate authentication
- gNMI JWT authentication
- gNMI password authentication
- View gNMI authentication
- gNMI request examples
- gNMI for streaming telemetry
- gRPC network operations interface

## gNMI certificate authentication

gNMI remote procedure calls require that you authenticate your access to a switch using one of the methods described in [gNMI Network Management Interface](#).

- (i) NOTE:** During an Enterprise SONiC upgrade or downgrade, locally configured users and their passwords and roles are properly migrated when installing a SONiC image using the `image install` command. The config migration scripts automatically migrate the `config_db.json`, `/etc/passwd`, `/etc/group`, `/etc/shadow`, `/etc/gshadow`, `/home/*`, and `/etc/sonic/cert/` directories, and `/var/spool/mail` files. However, if you reinstall Enterprise SONiC from ONIE, and manually migrate a configuration from one switch to another by copying and restoring a `config_db.json` file or by provisioning Enterprise SONiC using custom ZTP scripts, you must:
- Manually re-create or restore the certificate and private key files that are used for the REST and/or gNMI telemetry servers.
  - Manually reconfigure the local users using the `username password role` command or programmatic interfaces. Remotely authenticated users whose credentials are authenticated by RADIUS, TACACS+, or LDAP are not affected.

## Install a CA certificate for client certificate authentication

gNMI uses client certificate authentication, which requires a client certificate to be sent by the client that accesses the switch. An installed CA certificate validates each client certificate. A client certificate must be signed by a certificate authority (CA) installed in the trust store and contain the common name (CN) field set to the name of the user. To download and use a CA certificate for gNMI certificate authentication, follow these steps:

1. Install a CA certificate from the specified URL, where `certificate-url` is in one of these formats:
  - `ftp://userid:passwd@hostip/filepath` — Installs a host certificate file from a remote FTP server.
  - `home://filename` — Installs a host certificate file from the home directory.
  - `http://hostip/filepath` — Installs a host certificate file from a remote HTTP server.
  - `scp://userid:passwd@hostip/filepath` — Installs a host certificate file from a remote SCP server.The CA certificate is maintained across image upgrades. Installing a CA certificate triggers a certificate expiration check. To delete an installed CA certificate, use the `crypto ca-cert delete` command.

```
sonic# crypto ca-cert install cert-file certificate-url
```

```
sonic# crypto ca-cert install home://ca.crt
Processing certificate ...
Installed Root CA certificate as "ca"
CommonName = localhost
IssuerName = localhost
```

- (i) NOTE:** An alternative way to install the CA certificate is to enter only the first part of the command `crypto ca-cert install` and then press Enter. When prompted, paste the raw ASCII format of the certificate between the BEGIN

CERTIFICATE and END CERTIFICATE headers. To verify the installed CA certificate, use the `show crypto ca-cert file` command. For example:

```
sonic# crypto ca-cert install home://ca.crt
Processing certificate ...
Installed Root CA certificate as "ca"
CommonName = localhost
IssuerName = localhost
sonic# crypto cert delete all
sonic#
sonic# crypto
sonic# crypto ca-cert delete all
sonic# crypto ca-cert install
Certificate base file name: ca.crt
Paste certificate below.
Include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- headers.
Enter a blank line to abort this command.
Certificate:
-----BEGIN CERTIFICATE-----
MIIDRTCCAI2gAwIBAgIUFN1IQV72x5qbEgfVxi9T66SL54kwDQYJKoZIhvcNAQEL
BQAwFDESMBAGA1UEAwwJbG9jYWxob3N0MB4XDTIzMDIyMTE3NTEzN1oXDTMzMDIx
ODE3NTEzN1owFDESMBAGA1UEAwwJbG9jYWxob3N0MIIBIjANBgkqhkiG9w0BAQEFA
AAOCAQ8AMIIIBCgKCAQEAP6cstTaX8FHPtMEgBxVncB2YcWLpJyuKxiNlubVtjIjP
BhTHR4O2a01b380RInntrEI41Go5kOMaBB9eMs6XUIT+GbltxRwV9j4cSYvNcz0
i89KogoN59q7325i1iC2T/a+qs1XLtqPR5HvP1BfY8qX97vPZKv/Sd4iRaIrsqBq
tDgHkSPcUhevO/JG9jFh1jA/vTAAnRZaTbS2JwzILkadggIiWitCUFI6K24NAkJuR
wcPEUWtJZbhoXIB2Y8jBbBd0k+uASTcDr9ZB0leyC32GBtvGFoLWzpuuQNz8DKlm
YuZjU4XwuEowlFnkIG+KtyNCrh6YvvvFUTOFYMaQIDAQABo4GOMIGLMAwGA1Ud
EwQFMAMBAf8wHQYDVROOBYEFI4b1JxDiooaDE+XsYBDyspsNDMSME8GA1UdIwRI
MEAfI4b1JxDiooaDE+XsYBDyspsNDMSoRikFjAUMRIwEAYDVQQDALsb2Nhbgkv
c3SCFBTZSEFe9seamxIH1cYvU+uki+eJMAsgA1UdDwQEAwIBBjANBgkqhkiG9w0B
AQsFAAOCAQEAMhkFXtFmzN9sg8dISJ8afKNGTjVqpwkaVKMKyBaNmRB9Rn3qWp8V
i7r06vsQc+WmNc9PDQtazPRE1pJqBP1pb9kMCfFNGwDF6p7GW6oTTtfPMLimrCL0
NVe31g8DiXiY5j2yT+0kdL6/h0+vM7VRjTVW9ODt+1IM/W5B2yTfxTU+J2Ok6oFc
eYs1bCF5ag6UOsKArqlsf600TcnA51FpqdA0dsS89G7bpZg/nMXD2fHUEts362aZ
43tW01MUQ45EsONDWctnWpunVI9jhzNqQZFhza30klHeTfgfmDdfjmlMRde81jrC
mXsd4aQfR93FAE+RemvpOW9E/7Mw8D3GoQ==
-----END CERTIFICATE-----
Processing certificate ...
Installed Root CA certificate as "ca"
CommonName = localhost
IssuerName = localhost
```

2. Create a trust store in which you store the CA certificates that are used to verify access to gNMI.

```
sonic# crypto trust-store trust-store-name trust-store trust-store-name
```

For example:

```
sonic(config)# crypto trust-store gnmits
```

3. Store a CA certificate in the trust store. Enter a certificate name installed with the `crypto cert install cert-file keyfile` command. Re-enter the command to add multiple CA certificates in the trust store.

```
sonic# crypto trust-store trust-store-name ca-cert certificate-name
```

For example:

```
sonic# configure terminal
sonic(config)# crypto trust-store gnmits ca-cert CA
```

4. Associate the trust store with a security profile used to authenticate gNMI clients.

```
sonic(config)# crypto security-profile profile-name trust-store trust-store-name
```

For example:

```
sonic(config)# crypto security-profile gnmiserver trust-store gnmits
```

5. (Optional) Configure security profile settings.

- Require gNMI to verify if the key used to authenticate a remote device is associated with a CA certificate or a client certificate. Enter True to ensure that the correct certificate/key pair is used to access the switch; enter False not to check whether authentication is performed in CA or client certificate mode. Default: False.

```
sonic(config)# crypto security-profile profile-name key-usage-check {True | False}
```

- Require gNMI to verify if the remote device name matches the name on the certificate that is used to authenticate the device. True verifies the device name; False does not perform a remote device name check. Default: False.

```
sonic(config)# crypto security-profile profile-name peer-name-check {True | False}
```

- Require immediate revocation of an installed certificate if the revocation check returns a valid response. True performs a certificate check; False does not use certificate revocation. Default: False.

```
sonic(config)# crypto security-profile profile-name revocation-check {True | False}
```

- Add a global Certificate Revocation List (CRL) Distribution Point (CDP) list to receive CRL updates in addition to the CDPs defined in installed certificates. For *crl-list*, enter a comma-separate list of the URLs for remote CDP servers in the format `http://host-ip/filepath`.

```
sonic(config)# crypto security-profile cdp-list profile-name crl-list
```

For example:

```
sonic(config)# crypto security-profile cdp-list myserver http://a.example.com/cdp,http://b.example.com/cdp
```

- Add a global Online Certificate Status Protocol (OSCP) responder list in addition to the responders defined in installed certificates. For *oscp-list*, enter a comma-separate list of the URLs for remote OSCP responder servers in the format `http://host-ip/filepath`.

```
sonic(config)# crypto security-profile oscp-list profile-name oscp-list
```

For example:

```
sonic(config)# crypto security-profile oscp-list myserver http://a.example.com/oscp,http://b.example.com/oscp
```

- Enable the security profile for the gNMI and Telemetry service. When the Telemetry server restarts, it uses the new certificate.

```
sonic(config)# ip telemetry security-profile profile-name
```

```
sonic(config)# ip telemetry security-profile gnmiserver
```

## Example: Configure gNMI certificate authentication

This example shows how to install a CA certificate, create a trust store with the CA certificate, and associate the CA certificate trust store with a security profile for gNMI authentication:

```
sonic# crypto ca-cert install home://CA.crt
Processing certificate ...
Installed Root CA certificate
CommonName = www.dell.com
IssuerName = www.dell.com

sonic# configure terminal
sonic(config)# crypto trust-store gnmits ca-cert CA
sonic(config)# crypto security-profile gnmiserver trust-store gnmits
```

To check the status of an installed certificate to see if it has expired, use the `crypto cert verify certificate-name expiry` command; for example, when you verify an installed self-signed certificate:

```
sonic# crypto cert verify CA expiry
Certificate is valid!
```

**(i) NOTE:** gRPC and REST API servers that perform certificate authentication require that your remote device has a certificate and private key pair.

**(i) NOTE:** When you store a CA certificate, the expiration date of the certificate is checked once a day. When the certificate expiration is within 30 days, a Syslog message is generated once a day. When the certificate expires in less than 14 days, Syslog warnings are generated. When the certificate expires, critical Syslog messages are generated.

From the remote device, access the gRPC service on a switch by specifying the client certificate-key pair and CA certificate file in a gNMI RPC. If you do not specify a CA certificate or if you are using self-signed host certificates, specify the `-insecure` option to disable certificate validation.

```
./gnmi_get -cert client.crt -key client.key -xpath /openconfig-interfaces:interfaces/
interface[name=Eth1/1]/ -target_addr switch-ip-address:8080 -ca CA.cert -target_name
admin
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
>
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Eth1/1"
 >
 >
>
encoding: JSON_IETF
```

A successful login and curl command execution returns a response:

```
== getResponse:
notification: <
 timestamp: 1582569532183928802
 prefix: <
 >
 update: <
 path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
>
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Eth1/1"
 >
 >
>
```

# gNMI JWT authentication

A JSON web token is used to authenticate a username and role when accessing the gNMI interface to configure or monitor a switch. You must first generate a token and then use it in a gNMI request. A JWT token expires after the expiration period stated in the response.

## Generate the JWT token

```
./gnmi_client -module Sonic -rpc authenticate -jsonin '{"username": "admin", "password": "sonicadmin"}' -insecure
Sonic Authenticate
{"Token":
 {"access_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwicm9sZXMiOl
 siYWRtaW4iLCJzdWRvIwiZG9ja2VyIl0sImV4cCI6MTU4MTUzNDk4NX0.ThySin-fSEoH86R7nitx9kv6DyPO0Mp2zV8MDZX_10A","type":"Bearer","expires_in":3600}}
```

## Use the JWT token in a gNMI Get request

```
./gnmi_get -xpath /openconfig-interfaces:interfaces/interface[name=Eth1/1] / -insecure
-target_addr localhost:8080 -jwt_token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwicm9sZXMiOl
 siYWRtaW4iLCJzdWRvIwiZG9ja2VyIl0sImV4cCI6MTU4MTUzNDk4NX0.ThySin-fSEoH86R7nitx9kv6DyPO0Mp2zV8MDZX_10A
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Eth1/1"
 >
>
encoding: JSON_IETF

== getResponse:
notification: <
 timestamp: 1581531461243037727
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Eth1/1"
 >
 >
 >
 val: <
 json_iwf_val: "{\"openconfig-interfaces:interface\": [{\"config\": {\"enabled\": true, \"mtu\": 9100, \"name\": \"Eth1/1\"}, \"name\": \"Eth1/1\", \"openconfig-if-ethernet:ethernet\": {\"state\": {\"port-speed\": \"openconfig-if-ethernet:SPEED_25GB\"}}, \"state\": {\"admin-status\": \"UP\", \"description\": \"\", \"enabled\": true, \"mtu\": 9100, \"name\": \"Eth1/1\", \"oper-status\": \"DOWN\"}, \"subinterfaces\": {\"subinterface\": [{\"index\": 0}]}}}]}"
 >
>
```

# gNMI password authentication

Password authentication for gNMI requests requires you to provide a username and password to access a switch.

## gNMI password authentication for get interface counters

```
./gnmi_get -insecure -xpath /openconfig-interfaces:interfaces/
interface[name=Management0]/state/counters -target_addr switch-ip-address:8080 -username
admin -password sonicadmin

== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Management0"
 >
 >
 elem: <
 name: "state"
 >
 elem: <
 name: "counters"
 >
>
encoding: JSON_IETF

== getResponse:
notification: <
 timestamp: 1580320160838720664
 prefix: <
 >
 update: <
 path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Management0"
 >
 >
 elem: <
 name: "state"
 >
 elem: <
 name: "counters"
 >
 >
 val: <
 json_ietf_val: "{\"openconfig-interfaces:counters\":{\"in-discards\":\"0\",
\"in-errors\":\"0\", \"in-multicast-pkts\":\"0\", \"in-octets\":\"489787\", \"in-pkts\":
\"7013\",
\"out-discards\":\"0\", \"out-errors\":\"0\", \"out-octets\":\"3073273\", \"out-pkts\":
\"3292\"}}"
 >
 >
>
```

# View gNMI authentication

To view the authentication modes configured for gNMI authentication:

```
sonic# show ip telemetry authentication

Telemetry Client Authentication Modes

client_auth: password,jwt
```

To display the settings used in gNMI authentication and for authenticating access to other Telemetry services:

```
sonic# show ip telemetry
Log level is not-set
JWT valid is 3600 seconds
JWT refresh is 900 seconds
Port is 443
Client authentication mode is password
Security profile gnmiserver
```

To display information on installed CA certificates:

```
sonic# show crypto ca-cert {certificate-name | all}

sonic# show crypto ca-cert CA
Certificate Name: CA
Certificate:
 Data:
 Version: 3 (0x2)
 Serial Number:
 64:f2:8f:46:09:68:73:34:46:d4:53:94:2c:fc:1f:12:06:6c:ee:8c
 Signature Algorithm: sha256WithRSAEncryption
 Issuer: C = US, ST = CA, L = Sacramento, O = Dell, OU = Networking, CN =
www.dell.com
 Validity
 Not Before: Jul 7 20:24:34 2022 GMT
 Not After : Jul 4 20:24:34 2032 GMT
 Subject: C = US, ST = CA, L = Sacramento, O = Dell, OU = Networking, CN =
www.dell.com
 Subject Public Key Info:
 Public Key Algorithm: rsaEncryption
 RSA Public-Key: (2048 bit)
 Modulus:
 00:ea:27:4f:1e:9f:03:97:3f:77:f1:48:6a:29:46:
 86:ef:d9:d1:ec:e1:d0:93:28:88:f6:36:47:66:21:
 00:92:d9:d3:65:8d:3c:bc:80:ac:8f:22:ee:cd:20:
 5e:ec:47:13:e1:a8:85:20:01:73:30:94:8c:6a:0a:
 73:93:3c:0a:e5:f6:b1:0a:f5:ad:c1:0b:e2:84:3c:
 a6:5b:5f:7d:b8:71:da:ab:46:88:03:e8:42:63:52:
 76:8f:1d:3e:1a:c1:d4:dc:55:75:b8:8f:af:03:19:
 95:d3:84:62:e9:f9:1b:6c:9d:ff:a7:f6:8e:32:2f:
 32:7f:5d:05:79:84:e3:f4:cd:78:59:49:80:18:a8:
 d6:c5:6d:54:2d:07:1a:16:e0:ce:15:f4:c6:2c:ae:
 f5:8f:a1:89:aa:37:79:ea:1a:75:dd:11:df:ee:8a:
 87:ed:4e:3c:3a:54:96:d4:13:75:a0:af:57:94:b0:
 d5:13:80:3d:de:54:9b:e2:56:6b:1e:a1:d2:fd:93:
 d2:3b:77:7c:b7:d8:e9:7d:d0:5d:a1:dc:89:0e:a6:
 c4:82:e6:fc:15:4c:6a:df:45:8c:ab:0b:07:cf:49:
 b7:53:ee:2a:84:5a:ac:29:09:7c:ee:5a:ff:94:19:
 e2:7b:3f:11:7b:f7:dd:5b:2a:da:99:98:59:b8:07:
 23:e9
 Exponent: 65537 (0x10001)
 X509v3 extensions:
 X509v3 Basic Constraints:
 CA:TRUE
 X509v3 Key Usage:
 Digital Signature, Non Repudiation, Key Encipherment, Certificate Sign
 Signature Algorithm: sha256WithRSAEncryption
 63:9a:e3:b4:b7:10:d7:fa:92:cd:04:7b:f4:6a:c3:5e:7c:11:
```

```
c6:e6:9e:eb:61:60:97:cf:ad:60:03:ab:89:0f:e2:f1:5b:94:
7f:dd:6e:21:e9:21:60:0a:8d:81:ec:7e:35:84:b6:97:5e:65:
07:c4:e2:44:7c:8e:8c:7e:20:94:29:6d:32:e7:dc:6c:70:5d:
64:ed:cc:3a:d8:83:11:84:26:ab:87:f2:ce:05:8f:29:d2:0f:
62:de:a2:d9:86:cb:d4:7f:1c:60:f8:38:e3:41:9b:13:6d:24:
b4:eb:0d:ac:8b:5f:96:d4:0a:5c:26:aa:20:d3:c0:f8:06:24:
84:3d:d0:1b:e0:33:99:49:5e:3c:f9:77:68:94:d7:f2:11:be:
39:41:2a:44:5d:9e:a6:b6:a7:02:14:f4:02:6f:f1:f8:71:a5:
c2:ac:94:39:5d:b6:68:7d:00:5a:e5:92:74:5c:f7:52:5e:2d:
6a:4f:a6:0c:a6:1b:c1:ff:a9:46:1f:3c:5e:a1:16:fa:72:55:
1b:84:d2:a8:25:b1:c8:f2:35:97:e0:02:2c:08:9c:e3:69:0e:
2e:0d:9c:f1:98:25:28:06:dc:57:59:9d:bb:48:97:02:63:16:
80:80:b9:1e:5d:13:10:a7:8a:1c:84:2d:aa:7d:ec:3e:67:a0:
14:b5:d9:6a
```

To display the security profiles used in gNMI authentication:

```
sonic# show crypto security-profile

Security Profile : gnmiserver

Key Usage Check : False
Peer Name Check: False
Revocation Check: False
Certificate Name: None
Trust Store: None
CDP Identifiers: http://a.example.com/cdp,http://b.example.com/cdp
OCSP Responders: http://a.example.com/ocsp,http://b.example.com/ocsp

Security Profile : gnmisp

Key Usage Check : False
Peer Name Check: False
Revocation Check: False
Certificate Name: CA
Trust Store: None
CDP Identifiers: None
OCSP Responders: None
```

To display the trust stores used for gNMI authentication:

```
sonic# show crypto trust-store

Trust Store : gnmits

CA Certificates Name: CA
```

## gNMI request examples

To create RPCs used in gNMI requests, you can download sample gRPC code — such as `gnmi_get` and `gnmi_set` — from [project-arlo / sonic-telemetry](#) on GitHub. See [gRPC Network Management Interface](#) for more information.

### Create user

```
./gnmi_set -insecure -username admin -password sonicadmin -update /openconfig-
system:system/aaa/authentication/users:@./user.json -target_addr switch-ip-address:8080

== getRequest user.json:
{
 "openconfig-system:user": [
 {
 "username": "testadmin",Emerald44
 "config": {
 "username": "testadmin",
 "password": "test",
 "password-hashed": "test",
 "ssh-key": "string",
 "role": "admin"
```

```
 }
]
}
```

## Delete user

```
gnmi_set -insecure -username admin -password sonicadmin -delete /openconfig-
system:system/aaa/authentication/users/user[username=testuser] -target_addr switch-ip-
address:8080
```

## Get interface counters

```
./gnmi_get -insecure -xpath /openconfig-interfaces:interfaces/
interface[name=Management0]/state/counters -target_addr switch-ip-address:8080 -username
admin -password sonicadmin
```

```
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Management0"
 >
>
elem: <
 name: "state"
>
elem: <
 name: "counters"
>
>
encoding: JSON_IETF
```

```
== getResponse:
notification: <
 timestamp: 1580320160838720664
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Management0"
 >
 >
 elem: <
 name: "state"
 >
 elem: <
 name: "counters"
 >
 >
 val: <
 json_ietf_val: "{\"openconfig-interfaces:counters\":{\"in-discards\":\"0\",
\"in-errors\":\"0\", \"in-multicast-pkts\":\"0\", \"in-octets\":\"489787\", \"in-pkts\":
\"7013\",
\"out-discards\":\"0\", \"out-errors\":\"0\", \"out-octets\":\"3073273\", \"out-pkts\":
\"3292\"}}\"
```

```
>
>
>
```

## Set interface MTU

In the gnmi\_set command, mtu.json specifies a local file that contains the text describing the MTU to be set on the Eth1/1 interface; for example: {"mtu": 9000}.

```
gnmi_set -update /openconfig-interfaces:interfaces/interface[name=Eth1/1]/config/
mtu:@mtu.json -target_addr switch-ip-address:8080 -username admin -password sonicadmin
-insecure
```

```
== setRequest:
prefix: <
>
update: <
 path: <
 elem: <
 name: "sonic-interfaces:interfaces"
>
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Eth1/1"
>
>
 elem: <
 name: "config"
>
 elem: <
 name: "mtu"
>
>
val: <
 json_ietf_val: "{\"mtu\": 9100}"
>
>
```

```
== getResponse:
prefix: <
>
response: <
 path: <
 elem: <
 name: "sonic-interfaces:interfaces"
>
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Eth1/1"
>
>
 elem: <
 name: "config"
>
 elem: <
 name: "mtu"
>
>
op: UPDATE
>
```

## Enable interface

In the gnmi\_set command, enabled.json specifies a local file that contains the text describing the data to be set; for example:{ "enabled": true}.

```
gnmi_set -update /openconfig-interfaces:interfaces/interface[name=Eth1/1]/config/
enabled:@enabled.json -target_addr switch-ip-address:8080 -username admin -password
sonicadmin -insecure

== setRequest:
prefix: <
>
update: <
 path: <
 elem: <
 name: "sonic-interfaces:interfaces"
 >
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Eth1/1"
 >
 >
 elem: <
 name: "config"
 >
 elem: <
 name: "enabled"
 >
 >
 val: <
 json_ietf_val: "{\"enabled\": true}"
 >
>

== getResponse:
prefix: <
>
response: <
 path: <
 elem: <
 name: "sonic-interfaces:interfaces"
 >
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Eth1/1"
 >
 >
 elem: <
 name: "config"
 >
 elem: <
 name: "enabled"
 >
 >
 op: UPDATE
>
```

## Create VLAN

In the gnmi\_set command, vlan.json specifies a local file that contains the text describing the data to be set:

```
{"openconfig-interfaces:interface":[{"config":{"name ":"Vlan2","name ":"Vlan2","state ":"UP","enabled":true,"mtu":9100,"name ":"Vlan2"}, "subinterfaces": {"subinterface":[{"index":0}]}}]}]
```

or

```
{ "openconfig-interfaces:interface": [{ "config": { "name": "Vlan2" }, "name": "Vlan2", "state": { "admin-status": "UP", "enabled": true, "mtu": 9100, "name": "Vlan2" }, "subinterfaces": { "subinterface": [{ "index": 0 }] } }] }
```

```
gnmi_set -update /openconfig-interfaces:interfaces/interface/:@vlan.json -target_addr switch-ip-address:8080 -username admin -password sonicadmin -insecure
```

```
== setRequest:
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
>
 elem: <
 name: "interface"
>
>
 val: <
 json_ietf_val: "{\"openconfig-interfaces:interface\": [{\"config\":{\"name\":\"Vlan2\"},\"name\":\"Vlan2\",\"state\":{\"admin-status\":\"UP\", \"enabled\":true,\"mtu\":9100,\"name\":\"Vlan2\"},\"subinterfaces\":{\"subinterface\": [{\"index\":0}]}}]}"
>
>
```

```
== getResponse:
prefix: <
>
response: <
 path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
>
 elem: <
 name: "interface"
>
>
 op: UPDATE
>
```

### Subscribe to interface oper-status

```
./gnmi_cli -insecure -logtostderr -address switch-ip-address:8080 -query_type s
-streaming_type
TARGET_DEFINED -q /openconfig-interfaces:interfaces/interface[name=Eth1/1]/state/oper-
status -target OC-YANG -with_user_pass
```

```

username: admin
password:
{
 "OC-YANG": {
 "openconfig-interfaces:interfaces": {
 "interface": {
 "Eth1/1": {
 "state": {
 "oper-status": "{\"openconfig-interfaces:oper-status\": \"DOWN\"}"
 }
 }
 }
 }
 }
}

```

### Capabilities request

```

./gnmi_cli -capabilities -insecure -address switch-ip-address:8080 -with_user_pass
username: admin
password:

supported_models: <
 name: "openconfig-acl"
 organization: "OpenConfig working group"
 version: "1.0.2"
>
supported_models: <
 name: "openconfig-routing-policy"
 organization: "OpenConfig working group"
 version: "3.1.1"
>
supported_models: <
 name: "openconfig-mLAG"
 organization: "OpenConfig working group"
 version: "1.0.2"
>
supported_models: <
 name: "openconfig-acl"
 organization: "OpenConfig working group"
 version: "1.0.2"
>
...

```

### Subscribe to interface LLDP updates in on\_change mode

```

./gnmi_cli -insecure -logtostderr -address switch-ip-address:8080 -query_type s
-streaming_type ON_CHANGE
-q /openconfig-lldp:lldp/interfaces/interface[name=Eth1/22] -target OC-YANG
-with_user_pass
username: admin
password:

```

### Subscribe to ACL updates in sample mode

```

gnmi_cli -insecure -logtostderr -address switch-ip-address:8080 -query_type s
-streaming_sample_interval 20 -streaming_type SAMPLE -q /openconfig-acl:acl/ -v 0
-target OC-YANG -with_user_pass
username: admin
password: {
 "OC-YANG": {
 "openconfig-acl:acl": "{}"
 }
}

```

### Subscribe to ACL updates in polling mode

```

gnmi_cli -insecure -logtostderr -address switch-ip-address:8080 -query_type p
-polling_interval 1s -count 5 -q /openconfig-acl:acl/ -v 0 -target OC-YANG
-with_user_pass
username: admin

```

```

password: {
 "OC-YANG": {
 "openconfig-acl:acl": "{}"
 }
}
{
 "OC-YANG": {
 "openconfig-acl:acl": "{}"
 }
}
{
 "OC-YANG": {
 "openconfig-acl:acl": "{}"
 }
}
...

```

### Subscribe to ACL updates in once mode

```

gnmi_cli -insecure -logtostderr -address switch-ip-address:8080 -query_type o -q /
openconfig-acl:acl/ -v 0 -target OC-YANG -with_user_pass
username: admin
password: {
 "OC-YANG": {
 "openconfig-acl:acl": "{}"
 }
}

```

### Retrieve all data — configuration, state, and operational

The following GetRequest to a RADIUS server retrieves configuration, state, and operational data.

```
gnmi_get -xpath '/openconfig-system:system/aaa/server-groups/server-group[name=RADIUS]/
servers/server[address=10.10.10.10]' -logtostderr -target_addr localhost:8080
```

```

== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-system:system"
 >
elem: <
 name: "aaa"
>
elem: <
 name: "server-groups"
>
elem: <
 name: "server-group"
 key: <
 key: "name"
 value: "RADIUS"
 >
>
elem: <
 name: "servers"
>
elem: <
 name: "server"
 key: <
 key: "address"
 value: "10.10.10.10"
 >
>
>
```

```

== getResponse:
notification: <
 timestamp: 1634340866464546744
prefix: <
```

```

>
update: <
path: <
 elem: <
 name: "openconfig-system:system"
 >
 elem: <
 name: "aaa"
 >
 elem: <
 name: "server-groups"
 >
 elem: <
 name: "server-group"
 key: <
 key: "name"
 value: "RADIUS"
 >
 >
 elem: <
 name: "servers"
 >
 elem: <
 name: "server"
 key: <
 key: "address"
 value: "10.10.10.10"
 >
 >
 >
 val: <
 json_ietf_val: "
{
 \"openconfig-system:server\":[
 {
 \"address\": \"10.10.10.10\",
 \"config\":{
 \"address\": \"10.10.10.10\",
 \"openconfig-system-ext:auth-type\": \"chap\",
 \"openconfig-system-ext:priority\":1
 },
 \"radius\":{
 \"config\":{
 \"openconfig-aaa-radius-ext:encrypted\":true,
 \"retransmit-attempts\":1,
 \"secret-key\": \"\"
 },
 \"state\":{
 \"counters\":{
 \"access-accepts\": \"2\",
 \"access-rejects\": \"1\",
 \"openconfig-aaa-radius-ext:access-requests\": \"3\"
 },
 \"retransmit-attempts\":1
 }
 },
 \"state\":{
 \"address\": \"10.10.10.10\",
 \"openconfig-system-ext:auth-type\": \"chap\",
 \"openconfig-system-ext:priority\":1
 },
 \"tacacs\":{
 \"config\":{
 \"openconfig-aaa-tacacs-ext:encrypted\":true,
 \"secret-key\": \"\"
 }
 }
 }
]
}
"
>

```

```
>
>
```

### Retrieve only configuration data

The following GetRequest to a RADIUS server retrieves only configuration data.

```
gnmi_get -xpath '/openconfig-system:system/aaa/server-groups/server-group[name=RADIUS]/servers/server[address=10.10.10.10]' --data_type CONFIG -logtostderr -target_addr localhost:8080
```

```
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-system:system"
>
 elem: <
 name: "aaa"
>
 elem: <
 name: "server-groups"
>
 elem: <
 name: "server-group"
 key: <
 key: "name"
 value: "RADIUS"
 >
 >
 elem: <
 name: "servers"
>
 elem: <
 name: "server"
 key: <
 key: "address"
 value: "10.10.10.10"
 >
 >
>
```

```
== getResponse:
notification: <
 timestamp: 1634340926214511357
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-system:system"
>
 elem: <
 name: "aaa"
>
 elem: <
 name: "server-groups"
>
 elem: <
 name: "server-group"
 key: <
 key: "name"
 value: "RADIUS"
 >
 >
 elem: <
 name: "servers"
>
 elem: <
 name: "server"
>
```

```

 key: <
 key: "address"
 value: "10.10.10.10"
 >
 >
 val: <
 json_ietf_val: "
{
 \"openconfig-system:server\": [
 {
 \"address\": \"10.10.10.10\",
 \"config\": {
 \"address\": \"10.10.10.10\",
 \"openconfig-system-ext:auth-type\": \"chap\",
 \"openconfig-system-ext:priority\": 1
 },
 \"radius\": {
 \"config\": {
 \"openconfig-aaa-radius-ext:encrypted\": true,
 \"retransmit-attempts\": 1,
 \"secret-key\": \"\"
 }
 },
 \"tacacs\": {
 \"config\": {
 \"openconfig-aaa-tacacs-ext:encrypted\": true,
 \"secret-key\": \"\"
 }
 }
 }
]
}
"
>
>
>

```

### Retrieve only state data

The following GetRequest to a RADIUS server retrieves only state data, which includes operational data.

```
gnmi_get -xpath '/openconfig-system:system/aaa/server-groups/server-group[name=RADIUS]/servers/server[address=10.10.10.10]' --data_type STATE -logtostderr -target_addr localhost:8080
```

```

== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-system:system"
 >
 elem: <
 name: "aaa"
 >
 elem: <
 name: "server-groups"
 >
 elem: <
 name: "server-group"
 key: <
 key: "name"
 value: "RADIUS"
 >
 >
 elem: <
 name: "servers"
 >
 elem: <
 name: "server"
 key: <

```

```

 key: "address"
 value: "10.10.10.10"
 >
>

== getResponse:
notification: <
 timestamp: 1634340940535508475
 prefix: <
 >
 update: <
 path: <
 elem: <
 name: "openconfig-system:system"
 >
 elem: <
 name: "aaa"
 >
 elem: <
 name: "server-groups"
 >
 elem: <
 name: "server-group"
 key: <
 key: "name"
 value: "RADIUS"
 >
 >
 elem: <
 name: "servers"
 >
 elem: <
 name: "server"
 key: <
 key: "address"
 value: "10.10.10.10"
 >
 >
 >
 val: <START HERE
 json_ietf_val: "
{
 \"openconfig-system:server\":[
 {
 \"address\": \"10.10.10.10\",
 \"radius\": {
 \"state\": {
 \"counters\": {
 \"access-accepts\": \"2\",
 \"access-rejects\": \"1\",
 \"openconfig-aaa-radius-ext:access-requests\": \"3\"
 },
 \"retransmit-attempts\": 1
 }
 },
 \"state\": {
 \"address\": \"10.10.10.10\",
 \"openconfig-system-ext:auth-type\": \"chap\",
 \"openconfig-system-ext:priority\": 1
 }
 }
]
}
"
>
>
>

```

#### Retrieve only operational data

The following GetRequest to a RADIUS server retrieves only operational data with counters.

```
gnmi_get -xpath '/openconfig-system:system/aaa/server-groups/server-group[name=RADIUS]/servers/server[address=10.10.10.10]' --data_type OPERATIONAL -insecure -logtostderr -target_addr localhost:808
```

```
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-system:system"
>
 elem: <
 name: "aaa"
>
 elem: <
 name: "server-groups"
>
 elem: <
 name: "server-group"
 key: <
 key: "name"
 value: "RADIUS"
 >
 >
 elem: <
 name: "servers"
>
 elem: <
 name: "server"
 key: <
 key: "address"
 value: "10.10.10.10"
 >
 >
>
```

```
== getResponse:
notification: <
 timestamp: 1634340952348637025
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-system:system"
>
 elem: <
 name: "aaa"
>
 elem: <
 name: "server-groups"
>
 elem: <
 name: "server-group"
 key: <
 key: "name"
 value: "RADIUS"
 >
 >
 elem: <
 name: "servers"
>
 elem: <
 name: "server"
 key: <
 key: "address"
 value: "10.10.10.10"
 >
 >
>
```

```

>
val: <
 json_ietf_val: "
{
 \"openconfig-system:server\": [
 {
 \"address\": \"10.10.10.10\",
 \"radius\": {
 \"state\": {
 \"counters\": {
 \"access-accepts\": \"2\",
 \"access-rejects\": \"1\",
 \"openconfig-aaa-radius-ext:access-requests\": \"3\"
 }
 }
 }
 }
]
}
"
>
>
>

```

## gNMI for streaming telemetry

Network health relies on performance monitoring and data collection for analysis and troubleshooting. Network data is often collected with SNMP and CLI commands using pull mode. In pull mode, a management device sends a get request and pulls data from a client.

As the number of objects in the network and the metrics grow, traditional methods limit network scaling and efficiency. Using multiple management systems further limits network scaling. The pull model increases the processing load on a switch by collecting all data even when there is no change.

Streaming telemetry provides an alternative method where data is continuously transmitted from network devices with efficient, incremental updates. Operators subscribe to the specific data that they need using well-defined sensor identifiers.

While SNMP management systems poll for data even if there is no change, streaming telemetry enables access to near real-time, model-driven, and analytics-ready data. It supports more effective network automation, traffic optimization, and preventative troubleshooting. For example, streaming telemetry reports packet drops or high utilization on links in real time. A network automation application can use this information to provision new paths and optimize traffic transmission across the network. The data is encoded using standard IETF JSON and streamed using Google Protocol RPC (gRPC) transport.

You can use gNMI telemetry in a gRPC framework to stream data to:

- Open-source external collectors, such as Telegraph
- Proprietary network collectors that you implement

### gNMI modes

A gNMI-based telemetry session can stream data in two modes:

- Dial-in mode — The switch initiates a session with one or more collector devices according to the sensor paths and destinations in a subscription.
- Dial-out mode — A collector initiates a session with the switch.

**(i)** **NOTE:** Enterprise SONiC supports gNMI-based streaming telemetry only in dial-in mode. Only dial-in mode is supported for YANG-based streaming.

### Subscribe requests

A gNMI client subscribes for updates to one or more paths in one of the following modes:

- `once` — The target creates the relevant update messages, transmits them, and closes the remote procedure call (RPC).
- `poll` — A client sends a subscribe request message to the target that contains a `poll` field with an empty poll message. The target responds with updates to the empty fields.
- `stream` — Operates in `ON_CHANGE`, `SAMPLE` or `TARGET_DEFINED` mode. In a Subscribe stream request, only some fields support `ON_CHANGE`; all fields support `SAMPLE`.
  - `ON_CHANGE` — The target sends an update when a field value changes.

- SAMPLE — The target sends periodic updates. In SAMPLE mode, the sampling period is specified in the Subscribe Request message. Each field has a minimum default sampling interval.
- TARGET\_DEFINED — The target sends ON\_CHANGE updates if they are supported by the gNMI path. If not, the target sends SAMPLE updates.

## gNMI for telemetry examples

- Use gNMI to specify a sampling interval for streaming telemetry data:

```
gnmi_cli -insecure -logtostderr -address 127.0.0.1:8080 -query_type s
-streaming_sample_interval 20 -streaming_type SAMPLE -q /openconfig-acl:acl/ -v 0
-target OC-YANG -with_user_pass
username: admin
password: {
 "OC-YANG": {
 "openconfig-acl:acl": "{}"
 }
}
```

- Use gNMI to specify a polling interval for retrieving telemetry data:

```
gnmi_cli -insecure -logtostderr -address 127.0.0.1:8080 -query_type p
-pollling_interval 1s -count 5 -q /openconfig-acl:acl/ -v 0 -target OC-YANG
-with_user_pass
username: admin
password: {
 "OC-YANG": {
 "openconfig-acl:acl": "{}"
 }
}
}
^CE0206 01:04:23.625717 407 gnmi_cli.go:185] sendQueryAndDisplay(ctx,
{Addrs:[127.0.0.1:8080] AddressChains:[] Target:OC-YANG Replica:0 UpdatesOnly:false
Queries:[["openconfig-acl:acl"] Type:poll Timeout:30s NotificationHandler:<nil>
ProtoHandler:<nil> Credentials:0xc00012ece0 TLS:0xd6f6c0 Extra:map[]
SubReq:<nil> Streaming_type:TARGET_DEFINED Streaming_sample_int:0 Heartbeat_int:0
Suppress_redundant:false}, &{PollingInterval:1s StreamingDuration:0s Count:1
CountExhausted:false Delimiter:/ Display:0x833930 DisplayPrefix: DisplayIndent:
DisplayType:group DisplayPeer:false Timestamp: DisplaySize:false Latency:false
ClientTypes:[gnmil]}):
 client.Poll(): client.Poll(): EOF
```

- Use gNMI to specify a single retrieval of telemetry data:

```
gnmi_cli -insecure -logtostderr -address 127.0.0.1:8080 -query_type o -q /
openconfig-acl:acl/ -v 0 -target OC-YANG -with_user_pass
username: admin
password: {
 "OC-YANG": {
 "openconfig-acl:acl": "{}"
 }
}
```

## Subscribe requests using wildcard paths

In a gNMI request for all subscription modes — once, poll, and stream, you can use a wildcard one or more times in a path to specify the elements to poll in a subtree. To retrieve data, use an asterisk (\*) as a wildcard in a gNMI key value. For example:

- To retrieve the configured MTUs on all interfaces:

```
/openconfig-interfaces:interfaces/interface[name=*]/config/mtu
```

- To retrieve the admin state of all subinterfaces on all interfaces:

```
/openconfig-interfaces:interfaces/interface[name=*/subinterfaces/
subinterface[index=*/config/enabled
```

- To retrieve the admin state of all subinterfaces on a specified interface:

```
/openconfig-interfaces:interfaces/interface[name=Ethernet0]/subinterfaces/
subinterface[index=*/config/enabled
```

- To retrieve the admin state of a specified subinterface ID on any interface:

```
/openconfig-interfaces:interfaces/interface[name=*/subinterfaces/
subinterface[index=100]/config/enabled
```

**(i) NOTE:** Wildcard paths are supported only in gNMI Subscribe requests. The GetResponse message does not include wildcards; it includes the expanded paths. The multi-level ellipse (...) wildcard is not supported in gNMI Subscription requests.

**(i) NOTE:** In release 4.0 and later, wildcards are not supported in:

- A path element, such as /openconfig-interfaces:interfaces/\*/config or /openconfig-interfaces:interfaces/.../mtu.
- Wildcard paths that do not point to OpenConfig YANG-modeled data on the switch.

### Inband telemetry query using Management VRF

You can use gNMI to retrieve telemetry data on inband data ports. In the following example, a gNMI query is used to retrieve streaming telemetry on the Management VRF using port channel 256.

```
sonic# show ip vrf mgmt
VRF-NAME INTERFACES

mgmt Management0
 PortChannel256

sonic# show ip interfaces
Flags: U-Unnumbered interface, A-Anycast IP

Interface IP address/mask VRF Admin/Oper Flags

Management0 100.94.130.20/24 mgmt up/up
Loopback0 192.168.1.1/32 mgmt up/up
Loopback1 172.16.1.1/32 mgmt up/up
PortChannel256 192.168.0.0/31 mgmt up/up
sonic#

sonic# show PortChannel summary
Flags(oper-status): D - Down U - Up (portchannel) P - Up in portchannel (members)

Group PortChannel Type Protocol Member Ports

256 PortChannel256 (U) Eth LACP Eth1/1(P)

sonic# show running-configuration interface PortChannel256
!
interface PortChannel256
 no shutdown
 ip vrf forwarding mgmt
 ip address 192.168.0.0/31

sonic# show ip arp vrf mgmt
Type: R - Remote Neighbor entries (EVPN or MLAG Separate IP)

Address Hardware address Interface Egress Interface Type Action

100.94.130.252 e4:f0:04:63:af:7b Management0 - Dynamic Fwd
```

```

100.94.130.254 00:01:e8:8b:44:71 Management0 - Dynamic Fwd
192.168.0.1 1c:72:1d:9e:29:79 PortChannel1256 - Dynamic Fwd

gnmi_get -insecure -username admin -password linuxadmin -xpath /openconfig-
interfaces:interfaces/interface[name=Ethernet0]/config -target_addr 192.168.0.0:8080
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Ethernet0"
 >
>
elem: <
 name: "config"
>
>
encoding: JSON_IETF

== getResponse:
notification: <
 timestamp: 1687452277808891038
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Ethernet0"
 >
 >
 elem: <
 name: "config"
 >
 >
 val: <
 json_ietf_val: "{\"openconfig-interfaces:config\":{\"name\":\"Ethernet0\",\"type\":\"iana-if-type:ethernetCsmacd\"}}"
 >
>
>

```

## gRPC network operations interface

The gRPC network operations interface (gNOI) defines a set of gRPC-based services to configure, monitor, and stream data from a switch.

To create RPCs used in gNOI requests, you can download sample gRPC code — such as `gnoi_client` — from [project-arlo / sonic-telemetry](#) on GitHub. For more information, see [gRPC Network Management Interface](#).

### gNOI RPC for show tech support command

```

./gnoi_client -module Sonic -rpc showtechsupport
-jsonin "{\"sonic-show-techsupport:input\":{\"date\":\"2020-01-01T00:00:00.000Z\"}}"
-insecure -username username -password password

```

### **gNOI RPC for copy config command**

```
admin@sonic:~$./gnoi_client -module Sonic -rpc copyConfig
-jsonin '{"sonic-config-mgmt:input": {"source": "running-configuration", "destination":
"startup-configuration"}}' -insecure
Sonic CopyConfig
{"sonic-config-mgmt:output":{"status-detail":"SUCCESS."}}
```

### **gNOI RPC for clear interface counters**

```
admin@sonic:~$./gnoi_client -module OpenconfigInterfacesPrivate -rpc ClearCounters
-jsonin '{"openconfig-interfaces-private:input": {"interface-param": "all"}}' -insecure
Sonic OpenconfigInterfacesPrivateClearCounters Client
{"openconfig-interfaces-private:input": {"interface-param": "all"}
input: <
 interface_param: "all"
>
{"output":{"status_detail":"Success: Cleared Counters"}}
```

# Using OpenConfig paths

For switch configuration, management, and monitoring, Enterprise SONiC uses OpenConfig data models. Different Management Framework interfaces — CLI, REST API, gNMI — access the same YANG-modeled data. The following examples show how to perform the same operation using the REST API, gNMI and CLI interfaces with OpenConfig data paths.

For a complete list of Enterprise SONiC YANG models, go to the [Dell Technologies Support](#) site, search for Enterprise SONiC Distribution, open the Drivers & Downloads page, select a software release, and download the **Enterprise SONiC OS datamodel** zip file.

## Example: View interface configuration with OpenConfig

Using the Rest API:

```
$ curl -k https://localhost/restconf/data/openconfig-interfaces:interfaces/
interface=Management0/config -u admin:sonicadmin
{"openconfig-interfaces:config":
 {"description":"Management0","enabled":true,"mtu":1500,"name":"Management0","type":"iana-if-type:ethernetCsmacd"}}
```

Using gNMI:

```
root@sonic:/# gnmi_get -insecure -username admin -password sonicadmin -xpath /openconfig-interfaces:interfaces/interface[name=Management0]/config -target_addr 127.0.0.1:8080
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Management0"
 >
 >
elem: <
 name: "config"
 >
>
encoding: JSON_IETF
== getResponse:
notification: <
 timestamp: 1678736704770797742
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-interfaces:interfaces"
 >
 elem: <
 name: "interface"
 key: <
 key: "name"
 value: "Management0"
 >
 >
 elem: <
 name: "config"
 >
 >
 >
```

```

 val: <
 json_ietf_val: "{\"openconfig-interfaces:config\":
 \\"description\\":\"Management0\",\\\"enabled\\\":true,\\\"mtu\\\":1500,\\\"name\\\":\"Management0\",
 \\\"type\\\":\\\"iana-if-type:ethernetCsmacd\\\"} }"
 >
>
>
```

Using the CLI:

```

sonic# show interface Management 0
Management0 is up, line protocol is up
Hardware is MGMT, address is 54:bf:64:f3:f7:c1
Description: Management0
IPV4 address is 100.94.152.17/24
Mode of IPV4 address assignment: DHCP
IPV6 address is fe80::56bf:64ff:fef3:f7c1/64
Mode of IPV6 address assignment: DHCP
IP MTU 1500 bytes
LineSpeed 1GB, Auto-negotiation True
Input statistics:
 304376 packets, 174017188 octets
 318484 Multicasts, 0 error, 8105 discarded
Output statistics:
 42732 packets, 10793951 octets
 0 error, 0 discarded
Time since last interface status change: 2d19h28m
```

### **Example: View software images with OpenConfig**

Using the REST API:

```

$ curl -k -u admin:sonicadmin https://100.94.152.17/restconf/data/openconfig-image-
management:image-management
{"openconfig-image-management:image-management":{"global":{"state":{"current":"SONiC-
OS-dell_sonic_4.x_share.751-ae03eaa20","next-boot":"SONiC-OS-dell_sonic_4.x_share.751-
ae03eaa20"}}, "images":{"image":[{"image-name":"SONiC-OS-dell_sonic_4.x_share.735-
a91b8424d","state":{"image-name":"SONiC-OS-dell_sonic_4.x_share.735-a91b8424d"}}, {"image-
name":"SONiC-OS-dell_sonic_4.x_share.751-ae03eaa20","state":{"image-name":"SONiC-OS-
dell_sonic_4.x_share.751-ae03eaa20"}]}]}}
```

Using gNMI:

```

admin@sonic:~$ docker exec -it telemetry bash
root@sonic:/# gnmi_get -insecure -username admin -password sonicadmin -xpath /openconfig-
image-management:image-management -target_addr 127.0.0.1:8080
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-image-management:image-management"
 >
>
encoding: JSON_IETF

== getResponse:
notification: <
 timestamp: 1678737118538128020
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-image-management:image-management"
 >
 >
val: <
 json_ietf_val: "{\"openconfig-
image-management:image-management\":[\"global\":{\"state\":{\"current\":\"SONiC-OS-
dell_sonic_4.x_share.751-ae03eaa20\"},\"next-boot\":\"SONiC-OS-dell_sonic_4.x_share.751-
ae03eaa20\"}],\"images\":[{\"image\":[{\"image-name\":\"SONiC-OS-dell_sonic_4.x_share.735-"}]
```

```
a91b8424d\", \"state\": {\"image-name\": \"SONiC-OS-dell_sonic_4.x_share.735-a91b8424d\"}},
{\"image-name\": \"SONiC-OS-dell_sonic_4.x_share.751-ae03eaa20\", \"state\": {\"image-
name\": \"SONiC-OS-dell_sonic_4.x_share.751-ae03eaa20\"}}]} }"
>
>
>
```

Using the CLI:

```
sonic# show image list
Current: SONiC-OS-dell_sonic_4.x_share.751-ae03eaa20
Next: SONiC-OS-dell_sonic_4.x_share.751-ae03eaa20
Available:
SONiC-OS-dell_sonic_4.x_share.735-a91b8424d
SONiC-OS-dell_sonic_4.x_share.751-ae03eaa20
```

### **Example: View system information with OpenConfig**

Using the REST API:

```
$ curl -k -u admin:sonicadmin https://100.94.152.17/restconf/data/openconfig-
system:system/state
{ "openconfig-system:state": { "boot-time": "1678493803000000000", "current-
datetime": "2023-03-13T19:59:02Z", "hostname": "sonic", "openconfig-system-deviation:intf-
naming-mode": "STANDARD_EXT" } }
```

Using gNMI:

```
root@sonic:/# gnmi_get -insecure -username admin -password sonicadmin -xpath /openconfig-
system:system/state -target_addr 127.0.0.1:8080
== getRequest:
prefix: <
>
path: <
 elem: <
 name: "openconfig-system:system"
 >
 elem: <
 name: "state"
 >
>
encoding: JSON_IETF

== getResponse:
notification: <
 timestamp: 1678737581004138134
prefix: <
>
update: <
 path: <
 elem: <
 name: "openconfig-system:system"
 >
 elem: <
 name: "state"
 >
>
 val: <

json_ietf_val: "{\"openconfig-system:state\": {\"boot-time\": \"1678493804000000000\",
\"current-datetime\": \"2023-03-13T19:59:41Z\", \"hostname\": \"sonic\", \"openconfig-system-
deviation:intf-naming-mode\": \"STANDARD_EXT\" } }"
>
>
>
```

Using the CLI:

| Attribute | Value/State |
|-----------|-------------|
|           |             |

```

Boot Time : 2d19h40m
CurrentDatetime : 2023-03-13T19:57:07Z
Hostname : sonic
```

# Basic troubleshooting

## Best practices

- View traffic end-to-end from the application's view point
- Deploy network management infrastructure rapidly, where needed, when needed, and on-demand
- Extend analysis beyond the network and watch traffic to and from your host
- Focus on real-time assessment and use trend analysis to backup your conclusions
- Emphasize *effective* over *absolute* — leverage management solutions that resolve your most common, most expensive problem quickly
- Address networking performance issues before you focus on the application performance
- Use methodologies and technologies that fit your network and needs
- Continuously monitor performance and availability as a baseline for system performance and setup up time to quickly separate network issues from application issues

## Diagnostic tools

### Gather diagnostic information

You can generate a collection of information about switch configuration, operation, and logs for troubleshooting purposes. This information is helpful to analyze and diagnose problems that occur during switch operation, and proactively monitor network operation to minimize downtime.

Enterprise SONiC gathers diagnostic information about system hardware, operation, and software configuration by default, and stores the output at:

```
/var/dump/sonic_dump_sonic_date_time.tar.gz
```

For example:

```
/var/dump/sonic_dump_sonic_20191118_221625.tar.gz
```

Use the `show tech-support` command to start the collection of troubleshooting information and store the collected system information in a .tar.gz file. You can run only one collection process at a time. To reduce the tech-support file size, specify the starting time from which information is collected using the `since` option.

```
sonic# show tech-support [since {date time | yesterday}]
```

Enter the date in the format `YYYY-MM-DD`, where:

- `YYYY` is the year, such as 2021
- `MM` is the number of the month (01 to 12)
- `DD` is the number of the day (01 to 31)

Enter the `since` time in the format `THH:MMM:SS[.ddd...] {Z | +hh:mm | -hh:mm}`, where:

- Enter `T` to identify that a time parameter follows
- `HH` is the hour (01 to 24)
- `MM` is the number of minutes (00 to 59)
- `SS` is the second (01 to 60)
- `.ddd...` is an optional decimal of the specified second (example, `.234`)
- `Z` indicates that there is no offset from the specified time)
- `+hh:mm` indicates the hours and minutes to be added to the specified time and date
- `-hh:mm` indicates the hours and minutes to be subtracted from the specified time and date

Enter the option since yesterday to collect data for technical support starting from yesterday.

For example:

```
sonic# show tech-support since 2020-03-05T07:10:00Z
```

```
sonic# show tech-support since 2020-03-05T07:10:00-07:20
```

```
sonic# show tech-support since yesterday
```

To view the tech-support data collection progress, use the `show tech-support status` command. When the data collection finishes, the file name containing the collected output in a compressed .tar.gz file is displayed.

```
sonic# show tech-support status
Status: In Progress (Collecting Counters-1) Total items collected: 18
```

```
sonic# show tech-support status
Status: Completed
File Name: /var/dump/sonic_dump_sonic_20191008_082312.tar.gz
```

**(i) NOTE:** After you decompress and extract files from the compressed .tar.gz file, most of the extracted files are in readable format. Larger extracted files such as log files, core files, and other files that contain a large amount of output (dump of all BGP tables) are compressed in gzip format. These larger files have a .gz file extension.

To cancel the tech-support data collection, use the `show tech-support cancel` command.

```
sonic# show tech-support cancel
%Info: Tech-support process canceled
```

#### Send tech-support data to a remote server and view extracted file contents

To transfer the `show tech-support .tar.gz` output file from a switch to a remote server and view its contents:

1. Log in to the server and access a directory on the server to which you have write access. The directory must have at least 50 MB of available space.

```
admin@sonic:~$ mkdir dump
admin@sonic:~$ cd dump
```

2. Copy the `show tech-support .tar.gz` file to the directory using a supported file transfer method. When successful, the .tar.gz file and its file size display.

```
admin@sonic:~$ scp admin@switch-ip-address:/var/dump/
sonic_dump_sonic_20200113_232351.tar.gz ./ admin@password: *****
sonic_dump_sonic_20200113_232351.tar.gz 100% 2183KB 2.1MB/s 00:00
```

3. Extract the contents of the tar.gz file to the server directory using the `tar xvzf` command (example, to the `dump` directory).

```
admin@sonic:~$ tar xvzf sonic_dump_sonic_20200113_232351.tar.gz
sonic_dump_sonic_20200113_232351/
sonic_dump_sonic_20200113_232351/generate_dump
sonic_dump_sonic_20200113_232351/proc/
sonic_dump_sonic_20200113_232351/proc/vmstat
sonic_dump_sonic_20200113_232351/proc/ioports
sonic_dump_sonic_20200113_232351/proc/partitions
sonic_dump_sonic_20200113_232351/proc/net/
sonic_dump_sonic_20200113_232351/proc/net/ip6_tables_matches
sonic_dump_sonic_20200113_232351/proc/net/unix
...
```

The `show tech-support .tar.gz` files are extracted in a directory tree. The tree is organized according to the type of information contained in the files. Some examples of the file categories for which subdirectories are created in the output file tree are:

- Log files — log directory

- Linux configuration files — etc directory
- Generic application dump output — dump directory
- Network hardware driver information — sai directory
- Detailed information about various processes — proc directory

**(i) NOTE:** Use this command to extract the .tar.gz file contents to a different directory.

```
admin@sonic:~$ tar xvzf filename.tar.gz -C /destination-directory-path
```

- Display the contents of the directory in which you extracted the .tar.gz file. Switch to the top level of the extracted directory tree. Display the subdirectories in the directory tree; for example, debugsh, dump, log, and so on.

```
admin@sonic:~$ ls -ld *
drwxr-sr-x 8 userid ncore 4096 Jan 13 15:23 sonic_dump_sonic_20200113_232351
-rw-r--r-- 1 userid ncore 2235129 Jan 13 15:32 sonic_dump_sonic_20200113_232351.tar.gz

admin@sonic:~$ cd sonic_dump_sonic_20200113_232351
userid@xenlogin-eqx-05:~/sonic/showtech/dump/sonic_dump_sonic_20200113_232351

admin@sonic:~$ ls -d *
debugsh dump etc generate_dump log proc sai
```

- View the contents of a subdirectory in the directory tree; for example, log. The subdirectory contains compressed .gz files.

```
admin@sonic:~$ ls -d log/*
log/auth.log.gz log/iccpd.log.gz log/stpd.log.gz
log/bgpd.log.gz log/kern.log.1.gz log/swss.rec.gz
log/btmp.gz log/kern.log.gz log/syslog.1.gz
log/cron.log.gz log/mcelog.gz log/syslog.gz
log/daemon.log.1.gz log/messages.1.gz log/system.journal.gz
log/daemon.log.gz log/messages.gz log/telemetry.log.gz
log/debug.1.gz log/natorch_debug.log.gz log/udlfd.log.gz
log/debug.gz log/neighborh_debug.log.gz log/user.log.gz
log/dpkg.log.gz log/routeorch_debug.log.gz log/wtmp.gz
log/dropmonitororch_debug.log.gz log/sairedis.rec.1.gz log/ztp.log.gz
log/fdborch_debug.log.gz log/sairedis.rec.gz
```

- In the extracted contents of the .tar.gz output file, you can find an index file in HTML format that specifies the functional collection groups and the files collected in each group. Use a web browser to read the index file, which specifies the collection groups and their files in the directory tree of the local device. To read the index file, enter the following URL in a web browser:

```
file:///destination-directory-path/sonic_dump_sonic_YYYYMMDD_HHMMSS/
techsupport_index.html
```

- Extract the contents of a .gz file (for example, log/iccpd.log.gz) using the gunzip command. Use a text editor or the cat command to view the contents of an extracted file; for example, iccpd.log.

```
admin@sonic:~$ cd log
admin@sonic:~$ gunzip iccpd.log.gz
admin@sonic:~$ ls -d *iccp*
iccpd.log

admin@sonic:~$ cat iccpd.log
Jan 13 23:02:37.468023 sonic NOTICE iccpd#iccpd: [ICCP_FSM.NOTICE] Start ICCP: warm
reboot no
Jan 13 23:10:34.580879 sonic NOTICE iccpd#iccpd: [ICCP_FSM.NOTICE] Start ICCP: warm
reboot no
```

### View tech-support data on an attached terminal

As an alternative to generating the show tech-support .tar.gz output file, you can collect and display diagnostic information from various show commands on a terminal used to connect to a switch by using the show tech-support terminal command.

**(i) NOTE:** When you use the show tech-support terminal command, no tar.gz output file is created. Using the show tech-support terminal command allows you to log in to a switch and view and capture technical support

output without having to wait for the show tech-support tar.gz to complete and to transfer the tech-support data to a remote device.

```
sonic# show tech-support terminal

***#:sonic::tech-support-terminal:## show clock
Mon 16 Oct 2023 04:54:26 PM UTC

***#:sonic::tech-support-terminal:## show version

Software Version : rel_dell_sonic_4.x_share.1273-d83912624
Product : Generic
Distribution : Debian 10.13
Kernel : 5.10.0-21-amd64
Config DB Version : version_4_2_1
Build Commit : d83912624
Build Date : Thu Oct 12 09:21:41 UTC 2023
Built By : dngnetbuild.svc@jenkinsworker-eqx-03
Platform : x86_64-dellemc_s5212f_c3538-r0
HwSKU : DellEMC-S5212f-P-25G
ASIC : broadcom
Hardware Version : A06
Serial Number : TH0VK93CCET0021800DZ
Uptime : 16:54:28 up 2 days, 21:39, 2 users, load average: 1.44, 1.83, 1.70
Mfg : Dell EMC

REPOSITORY TAG IMAGE ID SIZE
docker-database latest 8a121a2081f3 628MB
docker-database rel_dell_sonic_4.x_share.1273-d83912624 8a121a2081f3 628MB
...

***#:sonic::tech-support-terminal:## show uptime
2 days, 21 hours, 39 minutes

***#:sonic::tech-support-terminal:## show users
admin pts/0 2023-10-13 19:21 (100.64.53.225)
admin pts/1 2023-10-16 16:53 (100.64.53.225)

***#:sonic::tech-support-terminal:## show image list
Current: SONiC-OS-rel_dell_sonic_4.x_share.1273-d83912624
Next: SONiC-OS-rel_dell_sonic_4.x_share.1273-d83912624
Available:
SONiC-OS-rel_dell_sonic_4.1.x_share.236-a56ad2029
SONiC-OS-rel_dell_sonic_4.x_share.1273-d83912624

***#:sonic::tech-support-terminal:## show image patch list

Id Tag Date DependsOn

***#:sonic::tech-support-terminal:## show image status

Global operation status : GLOBAL_STATE_IDLE

***#:sonic::tech-support-terminal:## show system status detail
System is ready
Service-Name Service-Status App-Ready-Status Down-Reason Status-UpdateTime
...
...
```

## tcpdump

tcpdump is a common packet analyzer and is used to display TCP/IP and other packets being transmitted or received over a network.

### Topics:

- System log
- Chassis locator LED
- Port locator LED
- Ping
- Traceroute
- Enable core file generation
- Error disable recovery
- Using port LEDs
- Port up or down troubleshooting
- Physical link signal
- Investigating packet drops
- Configure packet drop counters
- Buffer thresholds to detect congestion
- Debug application for congestion and drops
- Isolate SONiC switch from network
- NAT troubleshooting
- Kernel dump
- Check memory usage
- View the reason for down interfaces
- System reboot reason
- Unreliable Loss of Signal
- Transceiver and cable diagnostics

## System log

The system log (Syslog) records event messages from all Docker containers. Syslog messages are captured using `rsyslog` and saved in the `/var/log/syslog` file. The system log is enabled by default. The Syslog contains significant system events:

- Alerts for memory, CPU, and disk partition usage when a threshold limit is exceeded.
- Security events in the audit log
- Events that indicate system operation and alarm conditions

### Monitor system and per-process memory

Because memory is a critical resource within the system, it is essential to monitor the memory usage at the system and per-process levels, and report the memory usage across the system. Monitoring memory usage helps to identify memory distribution across the system, spikes in memory allocation, and if there are any memory leaks in the process. Thresholds are defined at the per-process and system level.

- System memory: When system memory usage crosses a threshold, a Syslog message is generated with overall system memory usage and memory usage of all running processes, including process name, process ID, and the amount of memory used. The following thresholds are used for system memory:
  - INFO - 0% to 70% of system memory (NORMAL)
  - WARN - 70% to 80% of system memory
  - ALERT - 80% to 90% of system memory
  - CRITICAL - 90% to 100% of system memory

Memory usage of the resource is displayed on the console in the format: Process name, Process ID, and RSS ( physical memory).

```
- Dec 11 13:06:19.397949 sonic WARN system#state: System memory usage is above 70%,
Total: 15.6G, Free: 1.8G, Used: 10.8G, Buffers: 314.8M, Cached: 2.7G
- Dec 11 13:06:19.477884 sonic INFO system#state: MEM :: Name: orchagent, Pid:6269,
Rss:10.5M
- Dec 11 13:06:19.477951 sonic INFO system#state: MEM :: Name: ospfd, Pid:11029,
Rss:10.5M
- Dec 11 13:06:19.478011 sonic INFO system#state: MEM :: Name: redis-server,
Pid:1006, Rss:10.6M
- Dec 11 13:06:19.478060 sonic INFO system#state: MEM :: Name: zebra, Pid:9625,
Rss:11.3M
```

- Per-process memory: When memory usage of a system process crosses a threshold, a Syslog message is generated with the process name, process ID, and the amount of memory used. The following thresholds are used for per-process memory:

- INFO - 0% to 30% of system memory (NORMAL)
- WARN - 30% to 40% of system memory
- ALERT - 40% to 50% of system memory
- CRITICAL - 50% to 100% of system memory

```
- Dec 11 13:03:19.209233 sonic INFO system#state: Per process memory threshold exceeded for process rest_server[3781], threshold 30% of system memory 478.6M, current usage 538.2M
- Dec 11 13:03:19.242928 sonic INFO system#state: Per process memory threshold exceeded for process syncd[14083], threshold 30% of system memory 478.6M, current usage 515.3M
```

### **Monitor per-process CPU usage**

The system monitors the CPU usage of all the processes. When CPU usage crosses a threshold, a Syslog message is generated with the process name, process ID, and CPU usage time. Each CPU threshold is determined by the duration of the sampling interval in which a high CPU usage is detected for a process:

- INFO - 0% to 70% of CPU utilization (NORMAL)
- WARN - 70% to 80% of high CPU utilization
- ALERT - 80% to 90% of high CPU utilization
- CRITICAL - 90% to 100% of high CPU utilization

### **Monitor disk partition usage**

Disk partitions are used for storing log files, core dumps, debug information, application files, configuration files, and Enterprise SONiC images. Monitoring disk partition tracks disk partition usage. When disk partition usage crosses a threshold, a Syslog message is generated with the partition name and amounts of used, free, and total disk space. The following thresholds are used for disk partition usage:

- INFO - 0% to 70% of total partition size (NORMAL)
- WARN - 70% to 80% of total partition size
- ALERT - 80% to 90% of total partition size
- CRITICAL - 90% to 100% of total partition size

```
Nov 27 07:16:50.878011 sonic INFO system#state: DISK usage of '/' is above 8%, Total: 31.4G, Free: 20.1G, Used: 9.7G Nov 27 07:16:50.878849 sonic INFO system#state: DISK:: {'used': 38285312, 'free': 3890614272, 'mountpoint': '/var/log', 'total': 4160421888}
```

### **Monitor security events from Audit log**

The audit log records messages about possible security events on the switch and sends them to the system log — see [Audit log](#). Audit log messages include:

- Logins and logouts from SSH and the console
- Configuration changes to a switch using MF-CLI configuration commands, the REST API, and gNMI
- Display of switch configurations using MF-CLI show commands, the REST API, and gNMI

Audit log messages are included in the `show techsupport` output. To display the twenty most recent audit log messages, use the `show audit-log` command. To clear all messages in the audit log, use the `clear audit-log` command.

**(i) NOTE:** The `show audit-log all` command displays all audit log messages and may impact switch performance due to the length of the output.

For example, a successful login message includes username and IP address from where the user logs in. Using SSH, the user admin logs in from a specified IP address and port with the timestamp when the login occurred:

```
Jun 2 22:47:08.619590 sonic INFO sshd[13990]: Accepted password for admin
from 10.14.8.140 port 49074 ssh2 Jun 2 22:47:08.711691 sonic INFO sshd[13990]:
pam_unix(sshd:session): session opened for user admin by (uid=0)
```

A successful login from the console displays as:

```
Jun 2 22:48:47.939333 sonic INFO login[30983]: Accepted password for admin
on terminal='/dev/ttyS0' Jun 2 22:48:48.056522 sonic INFO login[30983]:
pam_unix(login:session): session opened for user admin by LOGIN(uid=0)
```

An SSH login with an invalid username:

```
Jun 2 22:51:53.619712 sonic INFO sshd[31688]: Invalid user adminxxxx from 10.14.8.140
port 49090
```

A console login with an invalid password:

```
Jun 2 22:54:54.938982 sonic NOTICE login[6927]: pam_unix(login:auth): authentication
failure; logname=LOGIN uid=0 euid=0 tty=/dev/ttys0 ruser= rhost= user=admin Jun 2
22:54:57.568058 sonic NOTICE login[6927]: FAILED LOGIN (1) on '/dev/ttys0' FOR 'admin',
Authentication failure
```

Session timeout messages:

```
SSH Jun 10 19:26:32.887528 sonic INFO sshd[24578]: Timeout, client not responding.
Jun 10 19:26:33.025597 sonic INFO sshd[24481]: pam_unix(sshd:session): session closed
for user admin
```

```
Jun 10 20:02:33.904878 sonic INFO systemd[1]: Stopped Serial Getty on ttys0.
```

Configuration command entry from the command-line interface:

```
Jun 2 22:57:09.060819 sonic INFO mgmt-framework#clish: User "admin" command "tacacs-
server key mykey" status - success
```

The command syntax and the user who entered the command are displayed in the message. When the same configuration command is sent in a Set request using the REST API or gNMI:

```
Jun 12 19:33:40.728039 sonic INFO mgmt-framework#/usr/sbin/
rest_server[711]: [REST-5] User "admin@10.14.125.28:55937" request
"PATCH /restconf/data/openconfig-system:system/aaa/server-groups/server-group=TACACS/
config/openconfig-system-ext:secret-key" status - 204
```

Show command entry from the command-line interface:

```
Jun 2 22:55:55.171404 sonic INFO mgmt-framework#clish: User "admin" command "show tacacs-
server global" status - success
```

When the same configuration command is sent in a Get request using the REST API or gNMI:

```
Jun 12 19:36:04.059130 sonic INFO mgmt-framework#/usr/sbin/rest_server[711]: [REST-7]
User "admin@10.14.125.28:55937" request "GET /restconf/data/openconfig-system:system/aaa/
server-groups/server-group=TACACS/config/openconfig-system-ext:secret-key" status - 200
```

## Monitor events for system operation and alarms

The system log consists of entries that indicate a change in the state of the system, which can impact the operation and health of Enterprise SONiC applications. Event messages are logged in the system log. The event messages include:

- Single-occurring events that indicate a significant system operation are logged only once; for example, such as power supply failure and faulty fan functioning. System operation events persist in the system log across reloads, including restoring factory defaults, fast reboot, power recycling, and software upgrades and downgrades. These single-occurring events are tagged with the keyword EVENT and displayed in show event output.
- Alarm conditions that can be corrected and cleared, such as temperature that exceed a specified threshold. These conditions are dynamic and stateful. Raised conditions can transition to CLEARED, ACK, and UNACK. An event is logged for each transition. Alarm conditions are tagged with the keyword ALARM and displayed in show event and show alarm output.

Each event entry is entered with a severity level by the Enterprise SONiC component that raises it: CRITICAL, MAJOR, MINOR, WARNING, and INFORMATIONAL.

While an alarm condition exists, an Enterprise SONiC application logs a separate event with the action that describes changes in alarm status:

- A RAISE event is logged when a fault condition is detected.
- A CLEAR event indicates that the application has recovered from the alarm condition.
- ACKNOWLEDGE is entered by a support engineer to show that he is aware of the alarm condition and does not consider the fault to be catastrophic.

- UNACKNOWLEDGE restores the alarm to RAISE status and updates the alarm statistics.

**(i) NOTE:** After a reboot — cold boot, system warm reboot, or fast boot — an event persists across the reboot. Alarms do not persist across a reboot. After a switch restart, applications check to see if a condition exists and raises a corresponding event or alarm. After a power reset, some events may not persist according to when the system last performed a save.

## Configure System log servers

By default, the system does not send Syslog messages to a remote server. You must manually configure the servers on which you want to save Syslog messages.

You can also configure an optional source interface so that the system uses the specified IP address as the source interface. If you do not specify a source interface, the system uses the IP interface address of the outbound interface as the source address. If the interface has more than one configured IP address, the system uses the primary IP address.

To configure a remote Syslog server to receive all logged messages or only a subset of single-occurring system and alarm events:

```
sonic(config)# logging server {hostname | ip-address | ipv6-address} [source-interface
interface-type] [remote-port port-number] [vrf vrf-name] [message-type type] [severity
level] [vrf vrf-name]
```

- *hostname* - Enter the hostname of a Syslog server.
- *ip-address* - Enter the IP address of the Syslog server.
- *ipv6-address* - Enter the IPv6 address of the Syslog server.
- *source-interface interface-type* - (Optional) Enter an Ethernet, loopback, management, port channel, or VLAN interface IP address to be used as the source interface when sending packets.
- *remote-port port-number* - (Optional) Enter the remote port number. The range is from 1 to 65535.
- *message-type type* - (Optional) Enter a message type: *log* to send all Syslog messages or *event* to send only messages that are tagged with the keywords EVENT and ALARM for system operation and alarms.
- *severity level* - (Optional) Enter the severity level of the logged messages to be sent to a Syslog server. Messages only with the specified and higher severity levels are sent. Messages with lower severity levels are not forwarded to remote servers.
  - To forward all Syslog messages to a remote server, set the severity level to the lowest level 0 *emerg*. The security levels of Syslog messages are *debug(7)*, *info(6)*, *notice(5)*, *warning(4)*, *error(3)*, *crit(2)*, *alert(1)*, and *emerg(0)*. The default severity level is *notice*.
  - For Event messages (*message-type event*), the severity level is ignored. Event messages of all severity levels are forwarded to a Syslog server.
- *vrf vrf-name* - (Optional) Enter the name of the VRF used to send Syslog messages.

### Examples: Syslog server configuration

To configure the system to send Syslog messages to the remote server IP address 10.59.142.126:

```
sonic(config)# logging server 10.59.142.126
```

To send Syslog messages to 10.59.143.28 with loopback 1 as the source interface and from VRF1:

```
sonic(config)# logging server 10.59.143.28 source-interface Loopback 1 vrf Vrf1
```

To send Syslog messages to 10.59.136.33 with Eth1/1 as the source interface:

```
sonic(config)# logging server 10.59.136.33 source-interface Eth1/1
```

To send only Syslog messages for system operation and alarm events to 10.59.143.28 with loopback 1 as the source interface and from VRF1:

```
sonic(config)# logging server message-type event 10.59.143.28 source-interface Loopback
1 vrf Vrf1
```

### View configured Syslog servers

```
sonic# show logging servers

```

| HOST          | PORT | SOURCE-INTERFACE | VRF   | MESSAGE-TYPE | SEVERITY |
|---------------|------|------------------|-------|--------------|----------|
| 10.59.136.33  | 514  | Loopback1        | -     | log          | notice   |
| 10.59.142.126 | 514  | -                | -     | log          | error    |
| 10.59.143.28  | 514  | Eth1/1/1         | Vrf-1 | log          | notice   |

## View Syslog messages

To view messages that are stored in the system log, use the `show logging` command.

```
sonic# show logging [count | lines [number] | servers | filter {level level | since date-time | type log-type}]
```

- `count` — Displays the number of logged messages.
- `lines [number]` — Enter the number of lines to display. The range is from 1 to 65535.
- `servers` — View the configured system log servers.
- `filter {level level | since date-time | type message-type}` - Filter the displayed logs using a specified and higher severity levels, a date/time in the format `month day hh:mm:ss`, or a message type.

**i|NOTE:** To filter Syslog content, use the `| grep` option with the `show logging` command.

### Examples: View Syslog messages

To display all Syslog messages:

```
sonic# show logging
May 11 16:43:07.853550 2021 sonic NOTICE admin: Running sonic-clear logging
May 16 02:13:53.107861 2021 sonic ERR pidof[30142]: can't get program name from /proc/30123/stat
May 17 13:24:44.587237 2021 sonic WARNING snmp#snmp-subagent [sonic_ax_impl] WARNING: Missing lldp_loc_man_addr from APPL DB
May 18 09:48:59.883892 2021 sonic ERR pidof[12624]: can't get program name from /proc/12611/stat
May 20 12:42:07.712024 2021 sonic NOTICE root: hello
```

To display specified Syslog messages:

```
sonic# show logging | grep portchannel
May 21 17:14:20.885341 2021 sonic NOTICE teamd#teammgrd: :- setLagAdminStatus: Received admin status PortChannel1 for portchannel up.
```

To display the number of logged messages.

```
sonic# show logging count
```

To display the first three logged messages:

```
sonic# show logging lines 3
May 17 13:24:44.587237 2021 sonic WARNING snmp#snmp-subagent [sonic_ax_impl] WARNING: Missing lldp_loc_man_addr from APPL DB
May 18 09:48:59.883892 2021 sonic ERR pidof[12624]: can't get program name from /proc/12611/stat
May 20 12:42:07.712024 2021 sonic NOTICE root: hello
```

## Insert a message into Syslog

To insert a message into the system log, use the `logger` command.

```
sonic# logger abcd
SUCCESS
sonic# show logging
May 20 15:11:39.102466 2021 sonic NOTICE root: Running sonic-clear logging
May 20 15:28:19.084258 2021 sonic NOTICE root: abcd
```

## Clear logged messages

To clear all Syslog messages:

```
sonic# clear logging
```

### View only Event and Alarm messages

To view only the messages for single-occurring system operation and alarm events, use the `show event` command.

```
sonic# show event [details | summary | severity level | start timestamp end timestamp | recent {5min|60min|24hr} | id event-id | from event-id to event-id]
```

- `detail` — Displays detailed event information.
- `summary` — Displays summary information of logged events, including a summary of severity levels.
- `severity level` — Displays information for events with the specified severity level: `critical`, `major`, `minor`, `warning`, or `informational`. The default is `warning`.
- `start timestamp end timestamp` — Displays the events that are logged between the specified times. Enter the `timestamp` in the format `yyyy-mm-hhTmm:ss:msZ`, where `yyyy` is a 4-digit year, `mm` is a 2-digit month, `hh` is a 2-digit hour, and `Tmm:ss:msZ` is the hour-second-millisecond in the timestamp.
- `recent {5min|60min|24hr}` — Displays the most recent events that are logged in the last 5 minutes, 60 minutes, or 24 hours.
- `id event-id` - Displays information for the specified event ID number in `show event` output.
- `from event-id to event-id` - Displays information for the events in the range of the specified event IDs in `show event` output.

 **NOTE:** You can also use the `| grep` option to filter `show event` output.

### Examples: View only Event messages

To display all Event messages for system operation and alarms:

```
sonic# show event

Id Action Severity Name Timestamp Description

1 RAISE WARNING PSU_REMOVED 2023-10-20T09:17:54.479Z PSU 2
2 RAISE WARNING PSU_REMOVED 2023-10-20T09:17:54.488Z PSU 3
3 RAISE WARNING FAN_REMOVED 2023-10-20T09:17:56.985Z PSU 2 FAN 1
4 - INFORMATIONAL SYSTEM_STATUS 2023-10-20T09:19:59.868Z System is ready
5 - INFORMATIONAL SYSTEM_STATUS 2023-10-20T09:19:59.925Z System is not ready - one or
 more services are not up
6 - INFORMATIONAL SYSTEM_STATUS 2023-10-20T09:23:02.802Z System is ready
```

To display detailed information about Event messages:

```
sonic# show event details

Event Details - 1

Id: 1
Action: RAISE
Severity: WARNING
Type: PSU_REMOVED
Timestamp: 2023-10-20T09:17:54.479Z
Description: PSU 2
Source: PSU 2

Event Details - 2

Id: 2
Action: RAISE
Severity: WARNING
Type: PSU_REMOVED
Timestamp: 2023-10-20T09:17:54.488Z
Description: PSU 3
Source: PSU 3

Event Details - 3

Id: 3
Action: RAISE
Severity: WARNING
Type: FAN_REMOVED
```

```

Timestamp: 2023-10-20T09:17:56.985Z
Description: PSU 2 FAN 1
Source: PSU 2 FAN 1

Event Details - 4

Id: 4
Action: -
Severity: INFORMATIONAL
Type: SYSTEM_STATUS
Timestamp: 2023-10-20T09:19:59.868Z
Description: System is ready
Source: system_status

Event Details - 5

Id: 5
Action: -
Severity: INFORMATIONAL
Type: SYSTEM_STATUS
Timestamp: 2023-10-20T09:19:59.925Z
Description: System is not ready - one or more services are not up
Source: system_status

Event Details - 6

Id: 6
Action: -
Severity: INFORMATIONAL
Type: SYSTEM_STATUS
Timestamp: 2023-10-20T09:23:02.802Z
Description: System is ready
Source: system_status

Event Details - 7

Id: 7
Action: ACKNOWLEDGE
Severity: WARNING
Type: PSU_REMOVED
Timestamp: 2023-10-20T09:53:47.425Z
Description: Alarm id 1 ACKNOWLEDGE.
Source: 1

```

To display a summary of Event messages:

```

sonic# show event summary
Event summary

Total: 6
Raised: 3
Acknowledged: 0
Cleared: 0

```

To display the Event messages with a specified severity:

```

sonic# show event severity warning

Id Action Severity Name Timestamp Description

1 RAISE WARNING PSU_REMOVED 2023-10-20T09:17:54.479Z PSU 2
2 RAISE WARNING PSU_REMOVED 2023-10-20T09:17:54.488Z PSU 3
3 RAISE WARNING FAN_REMOVED 2023-10-20T09:17:56.985Z PSU 2 FAN 1
7 ACKNOWLEDGE WARNING PSU_REMOVED 2023-10-20T09:53:47.425Z Alarm id 1 ACKNOWLEDGE.
```

To display the Event messages logged within the last 24 hours:

```

sonic# show event recent 24hr

Id Action Severity Name Timestamp Description

1 RAISE WARNING PSU_REMOVED 2023-10-20T09:17:54.479Z PSU 2
2 RAISE WARNING PSU_REMOVED 2023-10-20T09:17:54.488Z PSU 3
3 RAISE WARNING FAN_REMOVED 2023-10-20T09:17:56.985Z PSU 2 FAN 1
4 - INFORMATIONAL SYSTEM_STATUS 2023-10-20T09:19:59.868Z System is ready
```

```

5 - INFORMATIONAL SYSTEM_STATUS 2023-10-20T09:19:59.925Z System is not ready - one or more
6 - INFORMATIONAL SYSTEM_STATUS 2023-10-20T09:23:02.802Z services are not up
7 ACKNOWLEDGE WARNING PSU_Removed 2023-10-20T09:53:47.425Z System is ready
 2023-10-20T09:53:47.425Z Alarm id 1 ACKNOWLEDGE.

```

To display the Event messages that are logged with IDs 2 to 5:

```

sonic# show event from 2 to 5

Id Action Severity Name Timestamp Description

2 RAISE WARNING PSU_Removed 2023-10-20T09:17:54.488Z PSU 3
3 RAISE WARNING FAN_Removed 2023-10-20T09:17:56.985Z PSU 2 FAN 1
4 - INFORMATIONAL SYSTEM_Status 2023-10-20T09:19:59.868Z System is ready
5 - INFORMATIONAL SYSTEM_Status 2023-10-20T09:19:59.925Z System is not ready - one or more
 services are not up

```

To display the Event messages logged within a specified timestamp period:

```

sonic# show event start 2023-10-20T09:17:55.000Z end 2023-10-20T09:19:59.900Z

Id Action Severity Name Timestamp Description

3 RAISE WARNING FAN_Removed 2023-10-20T09:17:56.985Z PSU 2 FAN 1
4 - INFORMATIONAL SYSTEM_Status 2023-10-20T09:19:59.868Z System is ready

```

### Acknowledge a raised alarm event

Acknowledge an alarm with a RAISE action to show that you are aware of the fault and do not consider the alarm condition to be significant. The alarm is then removed from the count in alarm statistics.

```
sonic# alarm acknowledge event-id
```

To unacknowledge an alarm so that it is considered in alarm statistics and restored to RAISED status:

```
sonic# alarm unacknowledge event-id
```

## View alarms

To filter logged events so that you view only the alarms that can be corrected and cleared, use the `show alarm` command.

**i | NOTE:** By default, the `show alarm` output displays only active alarm events. Acknowledged alarms are not shown.

```

sonic# show alarm [acknowledged | all | detail | summary | severity level | start
timestamp end timestamp | recent {5min|1hr|1day} | id
event-id | from event-id to event-id]

```

- `acknowledged` — Displays only acknowledged alarms.
- `all` — Displays information about all logged alarms, including acknowledged alarms.
- `detail` — Displays detailed alarm information.
- `summary` — Displays summary information of logged alarms.
- `severity level` — Displays information for alarms with the specified severity level: `critical`, `major`, `minor`, `warning`, or `informational`. The default is `warning`.
- `start timestamp end timestamp` — Displays the alarms that are logged between the specified times. Enter the `timestamp` in the format `yyyy-mm-hhTmm:ss:msZ`, where `yyyy` is a 4-digit year, `mm` is a 2-digit month, `hh` is a 2-digit hour, and `Tmm:ss:msZ` is the hour-second-millisecond in the timestamp.
- `recent {5min|60min|24hr}` — Displays the most recent alarms that are logged in the last 5 minutes, hour, or day.
- `id event-id` — Displays information about the specified alarm ID.
- `from event-id to event-id` — Displays information for the alarms in the range of the specified event IDs in `show event log` output.

### Examples: View alarms

```

sonic# show alarm

Id Severity Name Timestamp Description

```

|   |         |             |                          |             |
|---|---------|-------------|--------------------------|-------------|
| 2 | WARNING | PSU_REMOVED | 2023-10-20T09:17:54.488Z | PSU 3       |
| 3 | WARNING | FAN_REMOVED | 2023-10-20T09:17:56.985Z | PSU 2 FAN 1 |

**i | NOTE:** Acknowledged alarms are not shown by default in show alarm output. Acknowledged alarms are retriggered after a reboot, fast-reboot, or power cycle and displayed in the show alarm output if the alarm condition still exists.

```
sonic# show alarm all

Id Severity Name Timestamp Description

1 WARNING PSU_REMOVED 2023-10-20T09:17:54.479Z PSU 2
2 WARNING PSU_REMOVED 2023-10-20T09:17:54.488Z PSU 3
3 WARNING FAN_REMOVED 2023-10-20T09:17:56.985Z PSU 2 FAN 1
```

```
sonic# show alarm id 10
Id: 10
Severity: WARNING
Type: PSU_VOLTAGE_STATUS
Timestamp: 2019-03-01T08:26:42.384Z
Description: PSU 2: voltage out of range, current voltage=11.0, valid range=[None, None].
Source: PSU 2
Acknowledged: False
Acknowledged time: -
```

```
sonic# show alarm detail

Alarm Details - 1

Id: 1
Severity: WARNING
Type: PSU_REMOVED
Timestamp: 2023-10-20T09:17:54.479Z
Description: PSU 2
Source: PSU 2
Acknowledged: True
Acknowledged time: 2023-10-20T09:53:47.425Z

Alarm Details - 2

Id: 2
Severity: WARNING
Type: PSU_REMOVED
Timestamp: 2023-10-20T09:17:54.488Z
Description: PSU 3
Source: PSU 3
Acknowledged: False
Acknowledged time: -

Alarm Details - 3

Id: 3
Severity: WARNING
Type: FAN_REMOVED
Timestamp: 2023-10-20T09:17:56.985Z
Description: PSU 2 FAN 1
Source: PSU 2 FAN 1
Acknowledged: False
Acknowledged time: -
```

```
sonic# show alarm from 2 to 3

Id Severity Name Timestamp Description

2 WARNING PSU_REMOVED 2023-10-20T09:17:54.488Z PSU 3
3 WARNING FAN_REMOVED 2023-10-20T09:17:56.985Z PSU 2 FAN 1
```

```
sonic# show alarm summary
Alarm summary

Total: 2
Critical: 0
```

```

Major: 0
Minor: 0
Warning: 2
Acknowledged: 1

```

On N-series and E-series switches, when the redundant PSU is not present, the system displays the `PSU_REMOVED` and `FAN_REMOVED` warnings in the `show alarm` command output.

```

sonic# show alarm

Id Severity Name Timestamp
Description

8 WARNING PSU_REMOVED 2023-10-26T11:01:44.172Z PSU 2
9 WARNING FAN_REMOVED 2023-10-26T11:01:44.505Z PSU 2
FAN 1
```

On the N3248PXE-ON and E3248PXE-ON switches, when an external power supply is not present, the system displays the `PSU_REMOVED` warning in the `show alarm` command output.

```

sonic# show alarm

Id Severity Name Timestamp
Description

771 WARNING PSU_REMOVED 2023-10-19T07:36:36.565Z PSU 3
```

## Audit log

To monitor user activity and configuration changes on the switch, display the audit log. The audit log is enabled by default, and is stored at `/var/log` on the switch. Only the `admin` role can view and clear the audit log.

Use the audit log to troubleshoot security concerns. The audit log records:

- User logins and logouts from an attached console or SSH sessions.
- Configuration and show commands run using the Management Framework CLI, gNMI, and REST API operations.

Audit log entries are saved locally and are not sent to a configured Syslog server by default. You can send audit log entries of severity level `info` (6) or higher to a configured Syslog server by using the `logging server` command — [System logs](#); for example:

```
sonic(config)# logging server 100.94.218.203 severity info
```

### View audit log

To display the last 50 or so lines of audit entries, use the `show audit-log` command. To display all entries in the audit log, use the `show audit-log all` command.

```
sonic# show audit-log [all]
```

```

sonic# show audit-log
Jun 30 21:30:11.510641 sonic INFO sshd[9034]: Accepted password for admin from
10.14.8.140 port 39608 ssh2
Jun 30 21:30:11.652451 sonic INFO sshd[9034]: pam_unix(sshd:session): session opened for
user admin by (uid=0)
Jun 30 21:30:23.145054 sonic INFO mgmt-framework#clish: User "admin" command "clear
audit-log" status - success
Jun 30 21:31:06.149488 sonic INFO mgmt-framework#clish: User "admin" command "show audit-
log" status - failure
Jun 30 21:31:09.309332 sonic INFO mgmt-framework#clish: User "admin" command "show audit-
log all" status - failure
Jul 1 15:28:16.081409 sonic INFO sshd[6843]: Accepted password for admin from
```

```

10.14.8.140 port 47018 ssh2
Jul 1 15:28:16.194728 sonic INFO sshd[6843]: pam_unix(sshd:session): session opened for user admin by (uid=0)
Jul 1 15:28:59.022286 sonic INFO login[23748]: pam_unix(login:session): session closed for user admin
Jul 1 15:28:59.143034 sonic INFO systemd[1]: Stopped Serial Getty on ttys0.
Jul 1 15:29:03.328292 sonic INFO login[9873]: pam_unix(login:session): session opened for user admin by LOGIN(uid=0)
Jul 1 15:30:09.275533 sonic INFO login[9873]: pam_unix(login:session): session closed for user admin
Jul 1 15:30:09.393562 sonic INFO systemd[1]: Stopped Serial Getty on ttys0.
Jul 1 15:30:19.179230 sonic INFO sshd[14289]: Accepted password for admin from 10.14.8.140 port 47022 ssh2
Jul 1 15:30:19.277708 sonic INFO sshd[14289]: pam_unix(sshd:session): session opened for user admin by (uid=0)
Jul 1 15:30:30.737089 sonic INFO mgmt-framework#clish: User "admin" command "show tacacs-server global" status - success
Jul 1 15:30:53.990147 sonic INFO mgmt-framework#clish: User "admin" command "show interface status" status - success
Jul 1 15:31:07.753105 sonic INFO mgmt-framework#clish: User "admin" command "show version" status - success
Jul 1 15:31:19.224596 sonic INFO mgmt-framework#clish: User "admin" command "show authentication" status - success
Jul 1 15:31:27.776912 sonic INFO mgmt-framework#clish: User "admin" command "show lldp" status - success
Jul 1 15:31:34.805050 sonic INFO mgmt-framework#clish: User "admin" command "show wred" status - success
...

```

- If you use the REST API, specify the path: /restconf/operations/sonic-auditlog:get-auditlog.
- If you use gNMI, specify the path: /sonic-auditlog:get-auditlog.

#### **Clear audit log**

To clear all entries in the audit log, use the `clear audit-log` command.

```
sonic# clear audit-log
```

- If you use the REST API, specify the path: /restconf/operations/sonic-auditlog:clear-auditlog.
- If you use gNMI, specify the path: /sonic-auditlog:clear-auditlog.

## **Subscribe to Event messages using gNMI**

To subscribe and receive logged Syslog Event messages as they occur, you can use gNMI calls.

#### **Example: Subscribe to all Event messages (system operation and alarms) using gNMI**

```
./gnmi_get -xpath /openconfig-system:system/alarms -logtostderr -target_addr 100.104.22.193:8080 -username admin -password admin123 -insecure -time_out 100s
```

#### **Example: Subscribe to only Event messages for alarms using gNMI**

```
./gnmi_get -xpath /openconfig-system:system/events -logtostderr -target_addr 100.104.22.193:8080 -username admin -password admin123 -insecure -time_out 100s
```

## **Chassis locator LED**

The chassis locator LED allows you to identify a switch in a multi-switch installation or data center.

The chassis locator LED is on the front panel of a switch and labeled as LOC or loc. If a switch does not have a LOC LED, another front panel LED serves as the locator LED. When turned on, the color of the locator LED is platform-specific.

By default, the chassis locator LED is turned off. To turn the locator LED on, enter the `locator-led chassis on` command.

```
sonic# locator-led chassis on
Success
```

- If the chassis locator LED does not light up, a `Failed` message displays.

```
sonic# locator-led chassis on
Failed
```

- If the chassis locator LED is not supported on a switch, a `Not Supported` message displays.

```
sonic# locator-led chassis on
Not supported
```

- To turn the chassis locator LED off, enter the `locator-led chassis off` command.

```
sonic# locator-led chassis off
Success
```

To turn the chassis locator LED on only for a specified time in minutes (1 to 120), enter the `locator-led chassis on timer minutes` command; for example:

```
sonic# locator-led chassis on timer 5
Success
Locator LED will be off after 5 minutes
```

- To turn the chassis locator LED on for an indefinite time, enter the `locator-led chassis on` command. To turn off the LED, enter the `off` version of the command:

```
sonic# locator-led chassis on
Success

sonic# locator-led chassis off
```

To display the currently lit color of the locator LED on, enter the `show locator-led chassis` command. The LED color is platform-specific, and the color may vary.

```
sonic# show locator-led chassis
State Color

on blue
```

- If the locator LED is turned off, no color (`off`) displays.

```
sonic# show locator-led chassis
State Color

off off
```

- If the locator LED is on and if the software cannot retrieve the locator LED color, `unknown` is displayed. The LED color is lit on the switch.

```
sonic# show locator-led chassis
State Color

off Unknown
Color may vary with respect to system states
```

## Port locator LED

The Port Locator LED allows you to identify ports that may have cabling errors. Using the command-line interface, you can cause an interface LED or multiple interface LEDs to blink at one-second intervals. A blinking interface LED turns off all other

interface LEDs on the switch so that a network manager can easily locate a miss-wired interface. The color of an interface LED is platform-specific.

```
sonic# interface port-locator [Eth slot/port[/breakout-port] | Eth interface-range]
[timer minutes]
```

- `interface port-locator` — Enable the port locator LED on all interfaces.
- (Optional) `Eth slot/port | Eth interface-range` — Enable the port locator LED only on a specified interface or a range of interfaces. For an interface range, enter `Eth slot/port[/breakout-port]-slot/port[/breakout-port] [,slot/port[/breakout-port]-slot/port[/breakout-port] ...]`.
- (Optional) `timer minutes` — Enable the port locator LED only for the specified time in minutes (1 to 20; default 5).

To disable the port locator LED on all or specified interfaces, enter the `no interface port-locator [Eth slot/port[/breakout-port] | Eth interface-range]` command.

#### Usage notes

When you enable the port locator LED, an interface LED no longer indicates link and traffic activity with a solid light.

- If a port uses two LEDs — one for link status and one for traffic activity — only the Link LED blinks for the port locator function. The Activity LED is turned off.
- If the port interface has `link up` status, the interface LED is solidly lit only if the port locator is not enabled. If the port locator is enabled, the interface LED blinks and does not light solid.
- On an interface with a `link down` status, the interface LED remains off.
- On a 100G port with 4x25G breakout interfaces, enabling the port locator lights up only the first breakout LED. You can enable and disable the port locator on individual breakout interfaces.
- Port-locator settings are not saved when you save the switch configuration.
- Port-locator settings are not displayed in `show running-config` output.
- All interface LEDs return to normal operation to show link and traffic activity after you perform a hard or soft switch reboot.

#### Examples

To enable the port locator LED on all interfaces with a 5 minute timer:

```
sonic# interface port-locator timer 5
```

To enable the port locator LED on a range of interfaces and then disable it on specified interfaces:

```
sonic# interface port-locator Eth1/48-1/56
sonic# no interface port-locator Eth1/50,Eth1/54
```

To view the interfaces on which the port locator LED is enabled:

```
sonic# show interface port-locator

Interface Locator Mode Expiration Time
----- -----
Eth1/1 Disabled
Eth1/2 Disabled
Eth1/3 Disabled
Eth1/4 Enabled
Eth1/5 Enabled
Eth1/6 Enabled
Eth1/7 Enabled 2021-08-13 02:52:46
```

## Ping

To verify network connectivity and the accessibility of IPv4 and IPv6 devices, use the `ping` and `ping6` commands.

A ping command sends an Echo message and waits for a reply from the remote device for a specified timeout period. The Echo reply indicates that the remote device is active and contains the round-trip time, number of bytes sent, and time-to-live (TTL) value. The TTL is the number of hops back from the source to the destination.

 **NOTE:** The syntaxes for the `ping` and `ping6` commands contain the most commonly used options. For a complete parameter list, see the Linux man page of the `ping` command.

## Ping a remote device

```
{ping | ping6} [vrf {mgmt | vrf-name}] [-LRUbdfnqrVAB] [-c count] [-i interval] [-I interface] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos] [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline] [-W timeout] [hop1 hop2 ...] destination-address
```

To ping a remote device, enter the IPv4/IPv6 destination address and these optional values:

- **vrf {mgmt | vrf-name}** — (Optional) Pings an IPv4/IPv6 address in the management or a specified VRF instance.
- **-a** — (Optional) Audible ping.
- **-A** — (Optional) Adaptive ping. An interpacket interval adapts to the round-trip time so that one (or more, if you set the *preload* option) unanswered probe is present in the network. The minimum interval is 200 milliseconds for a nonsuper user, which corresponds to Flood mode on a network with a low round-trip time.
- **-b** — (Optional) Pings a broadcast address.
- **-B** — (Optional) Does not allow ping to change the source address of probes. The source address is bound to the address used when the ping starts.
- **-c count** — (Optional) Stops the ping after sending the specified number of ECHO\_REQUEST packets.
- **-d** — (Optional) Sets the SO\_DEBUG option on the socket being used.
- **-D** — (Optional) Prints the timestamp before each line.
- **-f** — (Optional) Flood ping. For every ECHO\_REQUEST sent, a period "!" is printed. For every ECHO\_REPLY received, a backspace is printed to provide a rapid display of how many packets are dropped.
- **-F flow-label** — (Optional - for ping6 only) Allocates and sets a 20-bit flow label on echo request packets. If the *flow-label* value is zero, the kernel allocates a random flow label.
- **-i interval** — (Optional) Enter the interval in seconds to wait between sending each packet, from 0 to 60; the default is 1 second.
- **-I interface** — (Optional) Enter the source interface *interface-type interface-number* without spaces or the interface IPv4/IPv6 address.
  - For a physical Ethernet interface, enter *Eth slot/port[/breakout-port]* (such as *Eth1/2/1*).
  - For a port channel interface, enter *PortChannelportchannel-number* (such as *PortChannel11*).
  - For a VLAN interface, enter *Vlanvlan-id* (such as *Vlan10*).
  - For a Loopback interface, enter *Loopbacknumber* (such as *Loopback0*).
  - For the Management interface, enter *Management0*.
- **-l preload** — (Optional) Enter the number of packets that ping sends before waiting for a reply. Only a superuser can pre-load more than three.
- **-L** — (Optional) Suppress the loopback of multicast packets for a multicast target address.
- **-m mark** — (Optional) Tags the packets sent to ping a remote device. Use this option with policy routing.
- **-M pmtudisc\_option** — (Optional) Enter the path MTU (PMTU) discovery strategy:
  - do prevents fragmentation, including local.
  - want performs PMTU discovery and fragments large packets locally.
  - dont does not set the Do not Fragment (DF) flag.
- **-p pattern** — (Optional) Enter a maximum of 16 pad bytes to fill out the packet you send to diagnose data-related problems in the network; for example, **-p ff** fills the sent packet with all 1's.
- **-Q tos** — (Optional) Enter a maximum of 1500 bytes in decimal or hexadecimal datagrams to set Quality of Service (QoS)-related bits.
- **-s packetsize** — (Optional) Enter the number of data bytes to send (1 to 65468; default 56).
- **-S sndbuf** — (Optional) Sets the sndbuf socket. By default, the sndbuf socket buffers one packet maximum.
- **-t ttl** — (Optional) Enter the IPv4/IPv6 time-to-live (TTL) value in seconds.
- **-T timestamp option** — (Optional) Set special IP timestamp options. Valid values for the *timestamp option* are *tsonly* (only timestamps), *tsandaddr* (timestamps and addresses), or *tsprespec host1 [host2 [host3 [host4]]]* (timestamp prespecified IPv4/IPv6 hops).
- **-v** — (Optional) Verbose output.
- **-V** — (Optional) Display the version and exit.
- **-w deadline** — (Optional) Enter the time-out value in seconds before the ping exits regardless of how many packets send or receive.
- **-W timeout** — (Optional) Enter the time to wait for a response in seconds. This setting affects the time-out only if there is no response, otherwise ping waits for two round-trip times (RTTs).
- **destination-address** — Enter the IPv4/IPv6 address of the remote device that you are trying to access.

## Ping examples

```
sonic# ping 20.1.1.1
PING 20.1.1.1 (20.1.1.1) 56(84) bytes of data.
64 bytes from 20.1.1.1: icmp_seq=1 ttl=64 time=0.079 ms
64 bytes from 20.1.1.1: icmp_seq=2 ttl=64 time=0.081 ms
64 bytes from 20.1.1.1: icmp_seq=3 ttl=64 time=0.133 ms
64 bytes from 20.1.1.1: icmp_seq=4 ttl=64 time=0.124 ms
^C
--- 20.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.079/0.104/0.133/0.025 ms
```

```
sonic# ping6 20::1
PING 20::1(20::1) 56 data bytes
64 bytes from 20::1: icmp_seq=1 ttl=64 time=2.07 ms
64 bytes from 20::1: icmp_seq=2 ttl=64 time=2.21 ms
64 bytes from 20::1: icmp_seq=3 ttl=64 time=2.37 ms
64 bytes from 20::1: icmp_seq=4 ttl=64 time=2.10 ms
^C
--- 20::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.078/2.194/2.379/0.127 ms
```

## Traceroute

To display the routes that packets take to a destination IPv4/IPv6 address, use the `traceroute` and `traceroute6` commands.

A traceroute command sends successive user datagram protocol (UDP) datagrams to a remote device with increasing TTL timeout values to 1, then 2, then 3, and so on, until the destination is reached. Each successive router in the path replies with an ICMP time exceeded message (TEM) to indicate that the destination has not been reached. The series of TEM messages trace the route to a destination device.

**i | NOTE:** The syntaxes for the `traceroute` and `traceroute6` commands contain the most commonly used options. For a complete parameter list, see the Linux man page of the `traceroute` command.

### Perform a traceroute

```
{traceroute | traceroute6} [vrf {mgmt | vrf-name}] [-46dFITnreAUDV] [-f first_ttl] [-g gateway,...] [-i interface] [-l flow_label] [-m max_ttl] [-N squeries] [-p port] [-P protocol] [-q nqueries] [-s src-address] [-t tos] [-UL] [-w waittime] [-z sendwait] [--mtu] [--back] host [packet_len]
```

To view the route path to a remote device, enter the IPv4/IPv6 destination host address and these optional values:

- `vrf {mgmt | vrf-name}`— (Optional) Traces the route to an IPv4/IPv6 address in the management or a specified VRF instance.
- `-f first_ttl`— (Optional) Enter the first time-to-live value to use. The default is 1.
- `-g gateway, ...`— (Optional) Adds an IP source routing option in the outgoing traceroute packet to route it through a specified gateway IPv4/IPv6 address. You can enter multiple, comma-separated gateway addresses. For IPv6, the form of `number, address, address...` is allowed, where `number` is a route header type (default is type 2). The Type 0 route header is deprecated (RFC 5095).
- `-i interface`— (Optional) Enter the interface through which traceroute sends packets. By default, the interface is selected using routing table entries. Enter `interface` in the format `interface-type interface-number` without spaces or enter the interface IPv4/IPv6 address.
  - For a physical Ethernet interface, enter `Eth slot/port[/breakout-port]` (such as `Eth1/2/1`).
  - For a port channel interface, enter `PortChannelportchannel-number` (such as `PortChannel1`).
  - For a VLAN interface, enter `Vlanvlan-id` (such as `Vlan10`).
  - For a Loopback interface, enter `Loopbacknumber` (such as `Loopback0`).
  - For the Management interface, enter `Management0`.
- `-l flow_label`— (Optional - for `traceroute6` only) Allocates and sets a 20-bit flow label on traceroute packets. If the `flow-label` value is zero, the kernel allocates a random flow label.

- **-m max\_ttl** — (Optional) Enter the maximum number of hops for the maximum time-to-live value that traceroute probes (default is 30).
- **-N squeries** — (Optional) Enter the number of probe packets sent out simultaneously to accelerate traceroute (default is 16).
- **-p port** — (Optional) Enter a destination port:
  - For UDP tracing, enter the destination port base that traceroute uses. The destination port number is incremented by each probe.
  - For ICMP tracing, enter the initial ICMP sequence value, which is incremented by each probe.
  - For TCP tracing, enter the constant destination port to connect.
- **-P protocol** — (Optional) Use a raw packet of the specified protocol for traceroute. The default protocol is 253 (RFC 3692).
- **-q nqueries** — (Optional) Enter the number of probe packets per hop (default is 3).
- **-s source\_address** — (Optional) Enter an alternative source address to use. By default, the IPv4/IPv6 address of the outgoing interface is used.
- **-t tos** — (Optional - for traceroute only) For IPv4, enter the type of service (ToS) and precedence values to use. 1.6 sets a low delay; 8 sets a high throughput.
- **-UL** — (Optional) Use UDPLITE for trace routing (default port is 53).
- **-w waittime** — (Optional) Enter the time in seconds to wait for a response to a probe (default is 5 seconds).
- **-z sendwait** — (Optional) Enter the minimal time interval to wait between probes. The default is 0. A value greater than 10 specifies a number in milliseconds, otherwise it specifies several seconds. This option is useful when routers rate-limit ICMP messages.
- **--mtu** — (Optional) Discovers the maximum transmission unit (MTU) from the path being traced.
- **--back** — (Optional) Prints the number of backward hops when different from the forward direction.
- **host** — (Required) Enter the name or IP address of the destination device.
- **packet\_len** — (Optional) Enter the total size (in bytes) of the probing packet (default is 60 for IPv4 and 80 for IPv6).

### Traceroute examples

```
sonic# traceroute www.dell.com
traceroute to www.dell.com (23.73.112.54), 30 hops max, 60 byte packets
1 10.11.97.254 (10.11.97.254) 4.298 ms 4.417 ms 4.398 ms
2 10.11.3.254 (10.11.3.254) 2.121 ms 2.326 ms 2.550 ms
3 10.11.27.254 (10.11.27.254) 2.233 ms 2.207 ms 2.391 ms
4 Host65.hbms.com (63.80.56.65) 3.583 ms 3.776 ms 3.757 ms
5 host33.30.198.65 (65.198.30.33) 3.758 ms 4.286 ms 4.221 ms
6 3.GigabitEthernet3-3.GW3.SCL2.ALTER.NET (152.179.99.173) 4.428 ms 2.593 ms 3.243 ms
7 0.xe-7-0-1.XL3.SJC7.ALTER.NET (152.63.48.254) 3.915 ms 3.603 ms 3.790 ms
8 TenGigE0-4-0-5.GW6.SJC7.ALTER.NET (152.63.49.254) 11.781 ms 10.600 ms 9.402 ms
9 23.73.112.54 (23.73.112.54) 3.606 ms 3.542 ms 3.773 ms

sonic# traceroute6 20:::1
traceroute to 20:::1 (20:::1), 30 hops max, 80 byte packets
1 20:::1 (20:::1) 2.622 ms 2.649 ms 2.964 ms
```

## Enable core file generation

Enable the generation of a core file when an application crash is detected by the kernel. Core file generation is enabled by default.

- Use this command in CONFIGURATION mode:

```
sonic(config)# core enable
```

## View core file information

- Check if the COREDUMP feature is administratively enabled or disabled:

```
show core config
```

## Example

```
sonic# show core config
Coredump : Enabled
sonic# show core config
Coredump : Disabled
```

- View detailed information about core problems:

```
show core info key
```

key — Process ID or executable name to match against.

## Example

```
sonic# show core info clish
Time : 2020-05-16 11:54:33
Executable : /usr/sbin/cli/clish
Core File : /var/lib/systemd/coredump/
core.clish.1000.8f1cad11c59840318a6df3aa6ed3633e.26480.158963007300000000
0000.1z4
PID : 26480
User ID : 1000
Group ID : 1000
Signal : 11
Command Line : /usr/sbin/cli/clish
Boot ID : 8f1cad11c59840318a6df3aa6ed3633e
Machine ID : fc0a437952314ee5a585a94ceaa480af
Core File Found : present
Crash Message :
Process 26480 (clish) of user 1000 dumped core.
Stack trace of thread 152:
#0 0x00007f30f516357a PyEval_EvalFrameEx (libpython2.7.so.1.0)
#1 0x00007f30f52cc29c PyEval_EvalCodeEx (libpython2.7.so.1.0)
#2 0x00007f30f5220670 n/a (libpython2.7.so.1.0)
#3 0x00007f30f51b85c3 PyObject_Call (libpython2.7.so.1.0)
#4 0x00007f30f52cb6c7 PyEval_CallObjectWithKeywords
(libpython2.7.so.1.0)
#5 0x00007f30eb2f43a n/a (/usr/sbin/cli/.libs/clish_plugin_clish.so)
```

- View a summary of the core files generated by the kernel:

```
show core list
```

## Example

```
sonic# show core list
TIME PID SIG COREFILE EXE
2020-05-16 11:54:33 26480 11 present clish
2020-05-15 01:25:16 6195 11 present crashme
2020-05-15 00:45:28 13604 11 present crashme
2020-05-14 02:11:11 3197 11 present crashme
2020-05-13 01:10:56 17844 11 missing crashme
2020-05-13 01:10:55 17728 11 present crashme
```

## Error disable recovery

You can configure the system to turn interfaces to the error disabled state for specific causes. You can also configure the system to automatically recover error-disabled interfaces after a set time.

- Enable error disable recovery for the specific cause:

```
errdisable recovery cause {[udld] | [bpduguard]}
```

- udld — (Optional) Enables a timer to recover from the unidirectional link detection (UDLD) error disabled state.
- bpduguard — (Optional) Enables a timer to recover from the bridge protocol data unit (BPDU) guard error disable state.

## Example

```
sonic(config)# errdisable recovery cause udld
sonic(config)# errdisable recovery cause bpguguard
```

- Configure the error disable recovery interval:

```
errdisable recovery interval interval
```

*interval* — Error disable recovery interval in seconds; default 300

- View the status of error disable recovery for all supported features:

```
show errdisable recovery
```

## Example

```
sonic# show errdisable recovery
ErrDisable Reason Timer Status

udld Disabled
bpduguard Disabled
```

## Using port LEDs

For port status, refer to the Port Activity and Link LEDs associated with a port. For the location of the LEDs, see the *Installation Guide* for a supported Enterprise SONiC switch.

**(i) NOTE:** On an Enterprise SONiC switch, the port LED behavior is different than on switches running OS10.

On an Enterprise SONiC switch, Port Activity and Link LEDs display as:

- Solid green — Link is up.
- Blinking green every 30 milliseconds — Port is transmitting traffic at any speed.
- Blinking green at one-second intervals — [Port locator LED](#).
- Off — Link is operationally down.

## Port up or down troubleshooting

This information describes how to determine if your network ports are up or down.

All port-related configuration that is done using the CLI or Config\_DB is saved in the redis configuration database. This configuration is handled by modules, and the result is stored in the application database (APP\_DB). If the modules complete their operation and if the result must be programmed into the ASIC, the result will be synchronized by the syncd service and stored in the ASIC\_DB

### Verify databases to debug or troubleshoot an issue

- Check the configuration in CONFIG\_DB, and the status using show commands.
- Check the application status of the application in the APP\_DB.
- Check the ASIC-related programming state, and the status in the ASIC\_DB.
- Check the actual ASIC.

### Check configuration and port status

- Check the interface status.

```
admin@sonic:~$ show interface status Ethernet1
Interface Lanes Speed MTU Alias Oper Admin

Ethernet1 101,102 40G 9100 Eth1/2 up up
```

- Check the interface transceiver presence.

```
admin@sonic:~$ show interfaces transceiver presence Ethernet1
Port Presence
```

```

Ethernet1 Present
```

3. Dump the port configuration from the ConfigDB.

```
admin@sonic:~$ redis-dump -d 4 -k "PORT|Ethernet1" -y
{
 "PORT|Eth1/2": {
 "type": "hash",
 "value": {
 "admin_status": "up",
 "alias": "Eth1/2",
 "description": "Servers1:Management0",
 "index": "2",
 "lanes": "53,54",
 "mtu": "9100",
 "pfc_asym": "off",
 "speed": "50000"
 }
 }
}
```

4. Check the port status in the APP\_DB.

```
admin@sonic:~$ redis-dump -d 0 -k *PORT_TABLE:Ethernet1* -y
{
 "PORT_TABLE:Ethernet1": {
 "type": "hash",
 "value": {
 "admin_status": "down",
 "alias": "Eth1/2",
 "description": "Eth1/2",
 "index": "16",
 "lanes": "95,96",
 "mtu": "9100",
 "oper_status": "down",
 "pfc_asym": "off",
 "speed": "50000"
 }
 }
}
```

5. Check the port status in the ASIC\_DB.

```
admin@sonic:~$ redis-dump -d 1 -k *ASIC_STATE* -y
"ASIC_STATE:SAI_OBJECT_TYPE_PORT:oid:0x1000000000014" -y
{
 "ASIC_STATE:SAI_OBJECT_TYPE_PORT:oid:0x1000000000014": {
 "type": "hash",
 "value": {
 "NULL": "NULL",
 "SAI_PORT_ATTR_ADMIN_STATE": "true",
 "SAI_PORT_ATTR_INGRESS_ACL": "oid:0xb000000000a61",
 "SAI_PORT_ATTR_MTU": "9122",
 "SAI_PORT_ATTR_PORT_VLAN_ID": "1000",
 "SAI_PORT_ATTR_PRIORITY_FLOW_CONTROL": "24",
 "SAI_PORT_ATTR_QOS_DSCP_TO_TC_MAP": "oid:0x14000000000a34",
 "SAI_PORT_ATTR_QOS_PFC_PRIORITY_TO_QUEUE_MAP": "oid:0x14000000000a35",
 "SAI_PORT_ATTR_QOS_TC_TO_PRIORITY_GROUP_MAP": "oid:0x14000000000a38",
 "SAI_PORT_ATTR_QOS_TC_TO_QUEUE_MAP": "oid:0x14000000000a39",
 "SAI_PORT_ATTR_SPEED": "50000"
 }
 }
}
```

6. Check the port status for the Broadcom ASIC. From the Linux shell, access the Broadcom shell. Press Ctrl+C to exit.

```
admin@sonic:~$ sudo bcmsh
BCM : bcmcmd ps
 port ena/link Lanes Speed Duplex LinkScan AutoNeg? STPstate pause discrd
LrnOps Int
 xe0(50) down 2 50G FD SW No Forward None FA
 KR2
 xe1(51) down 2 50G FD SW No Forward None FA
 KR2
```

|             |   |     |    |    |    |         |      |    |
|-------------|---|-----|----|----|----|---------|------|----|
| xe2( 54) up | 2 | 50G | FD | SW | No | Forward | None | FA |
| KR2         |   |     |    |    |    |         |      |    |

## Physical link signal

This information describes how to determine the optical signal strength.

**(i) NOTE:** Not all link types display signal strength values. For example, AOC cables have power values, but DAC cables do not have them.

Optical power should be greater than -10dBm.

```
root@sonic:/# redis-cli -n 6 hgetall "TRANSCEIVER_DOM_SENSOR|Ethernet1"
1) "temperature"
2) "0.0000"
3) "voltage"
4) "6.5280"
5) "rx1power"
6) "-28.2391"
7) "rx2power"
8) "N/A"
9) "rx3power"
10) "N/A"
11) "rx4power"
12) "N/A"
13) "tx1bias"
14) "8.7200"
15) "tx2bias"
16) "N/A"
17) "tx3bias"
18) "N/A"
19) "tx4bias"
20) "N/A"
21) "tx1power"
22) "-28.2391"
23) "tx2power"
24) "N/A"
25) "tx3power"
26) "N/A"
27) "tx4power"
28) "N/A"
29) "temphighalarm"
30) "75.0000"
31) "temphighwarning"
32) "70.0000"
33) "templowalarm"
34) "5.0000"
35) "templowwarning"
36) "10.0000"
37) "vcchighalarm"
38) "3.6300"
39) "vcchighwarning"
40) "3.4650"
41) "vcclowalarm"
42) "2.9700"
43) "vcclowwarning"
44) "3.1350"
45) "txpowerhighalarm"
46) "N/A"
47) "txpowerlowalarm"
48) "N/A"
49) "txpowerhighwarning"
50) "N/A"
51) "txpowerlowwarning"
52) "N/A"
53) "rxpowerhighalarm"
54) "4.9969"
55) "rxpowerlowalarm"
56) "-11.8977"
57) "rxpowerhighwarning"
```

```

58) "4.0000"
59) "rxpowerlowwarning"
60) "-7.8995"
61) "txbiashighalarm"
62) "10.0000"
63) "txbiaslowalarm"
64) "0.5000"
65) "txbiashighwarning"
66) "9.5000"
67) "txbiaslowwarning"
68) "1.0000"

```

## Investigating packet drops

This information describes how to investigate packet drops using the `show interfaces counters` command.

- RX\_ERR/TX\_ERR — includes all physical layer (L2) related drops such as FCS error and RUNT frames. If there is an RX\_ERR or TX\_ERR, it indicates some physical layer link issues.
- RX\_DRP — includes all L2, L3, ACL-related drops in the switch ingress pipeline, and drops due to insufficient ingress buffer.
- TX\_DRP — includes mainly the egress buffer related drop due to congestion including WRED drop.
- RX\_OVR/TX\_OVR — counts the oversized packets.

| sonic# show interface counters |                 |             |         |        |        |        |                 |        |  |
|--------------------------------|-----------------|-------------|---------|--------|--------|--------|-----------------|--------|--|
| Iface                          | RX_OK           | RX_RATE     | RX_UTIL | RX_ERR | RX_DRP | RX_OVR | TX_OK           | TX     |  |
| Eth1/2                         | 471,729,839,997 | 653.87 MB/s | 12.77%  | 0      | 18,682 | 0      | 409,682,385,925 | 556.84 |  |
| Eth1/3                         | 453,838,006,636 | 632.97 MB/s | 12.36%  | 0      | 1,636  | 0      | 388,299,875,056 | 529.34 |  |
| Eth1/4                         | 549,034,764,539 | 761.15 MB/s | 14.87%  | 0      | 18,274 | 0      | 457,603,227,659 | 615.20 |  |
| Eth1/5                         | 458,052,204,029 | 636.84 MB/s | 12.44%  | 0      | 17,614 | 0      | 388,341,776,615 | 527.37 |  |
| Eth1/6                         | 16,679,692,972  | 13.83 MB/s  | 0.27%   | 0      | 17,605 | 0      | 18,206,586,265  | 17.51  |  |
| Eth1/8                         | 47,983,339,172  | 35.89 MB/s  | 0.70%   | 0      | 2,174  | 0      | 58,986,354,359  | 51.83  |  |
| Eth1/9                         | 33,543,533,441  | 36.59 MB/s  | 0.71%   | 0      | 1,613  | 0      | 43,066,076,370  | 49.92  |  |

## Configure packet drop counters

**(i) NOTE:** Custom drop counter configuration is available only in the Cloud Standard, Cloud Premium, Enterprise Standard, and Enterprise Premium bundles. It is not available in the Edge Standard bundle.

To obtain detailed, customized views on packet drops and the corresponding drop reasons, you can configure mirroring sessions to mirror drop frames for further analysis.

Configuring packet drop counters at the port level allows you to count, classify, and capture packet drops that occur for different reasons. In addition, you can track the reasons why packets are dropped.

**(i) NOTE:** A mirroring session may not capture all dropped frames on the switch. This capability is ASIC-dependent.

### Create a custom drop filter

One use of packet-drop configuration is to create a drop filter to compare with the standard STAT\_IF\_IN/OUT\_DISCARDS counter count. For example, if packets X, Y, and Z exist in the system and the switch should drop them from incoming traffic:

- Create a new drop counter (for example, `Expected_Drops`) that counts X, Y, and Z packet drops.
- Compare the `Expected_Drops` number with the `STAT_IF_IN/OUT_DISCARDS` counter count.
- If the two numbers of dropped packets are the same, packet transmission is normal on the switch. However, if the `STAT_IF_IN/OUT_DISCARDS` number is greater, there may be an issue to troubleshoot on the switch.

### Debug packet loss issues

To troubleshoot packet loss on the switch, you can configure new packet drop counters. For example:

- Create two new drop counters: `L2_ANY` to track L2 packet drops and `L3_ANY` to track L3 packet drops.
- If `L2_ANY` drops increase, delete `L2_ANY` and `L3_ANY`. Create three new counters:
  - `MAC_COUNTER` to track MAC-related drop reasons, such as `SMAC_EQUALS_DM` and `DMAC_RESERVED`.
  - `VLAN_COUNTER` to track VLAN-related drop reasons, such as `INGRESS_VLAN_FILTER` and `VLAN_TAG_NOT_ALLOWED`.
  - `OTHER_COUNTER` to track all other drop reasons, such as `EXCEEDS_L2_MTU` and `FDB_UC_DISCARD`.

- If OTHER\_COUNTER increases, delete MAC\_COUNTER and VLAN\_COUNTER. Create a new counter to track the individual reasons for OTHER\_COUNTER drops.
- If packet drops increase for one of the OTHER\_COUNTER reasons, such as EXCEEDS\_L2\_MTU, there may be an MTU mismatch occurring on one or more switch ports.

### Sample specific types of packet drops

Using packet drop configuration settings, you can set up more sophisticated monitoring schemes. For example:

- Cycle through different drop counters on the switch at a specific time interval (for example, 30 seconds).
- Configure different types of packet drop monitoring on different switches. For example, configure three switches to monitor VLAN-related drops; configure three other switches to monitor ACL-related drops, and so on.
- Program a configuration command response to the increase of specified drop counters.

### Capture dropped frames

To aid in debugging, capture dropped frames and mirror them to the CPU or a switch port for analysis.

- Dropped frames that you mirror to the CPU are stored in the drop packet queue. From the Linux shell, use the Linux `drop_pkt_drop` command to display the captured frames. A maximum of 256 dropped frames are displayed from the dropped packet queue.
- Dropped frames that you mirror to a port can be forwarded in a SPAN or ERSPAN mirror session for further analysis — see [Port monitoring](#).

**(i) NOTE:** A mirroring session may not capture all dropped frames on the switch. This capability is ASIC-dependent. Some of the reasons for dropped frames in a mirror session are:

- UNKNOWN\_VLAN — The packet's VLAN is unknown.
- MARTIAN\_ADDR — The packet has the source IP address 0.0.0.0.
- DOS\_ATTACK — The packet's source IP address is the same as its destination IP address.
- L3\_MTU\_FAIL — The packet's MTU is greater than its configured MTU.
- L3\_ADDR\_BIND\_FAIL — The L3 packet has an incorrect source MAC address.
- INVALID\_TPID — The packet has an invalid tag protocol identifier (TPID).
- L3\_HEADER\_ERROR — There is an error in the packet's L3 header.
- TTL1 — The packet has a TTL value of 0.
- TTL — The packet has a TTL value of 1.

An Enterprise SONiC switch may support only a subset of these dropped frame reasons in a mirror session.

## Custom drop counter configuration

1. Display the default types of drop counters that you can monitor and the drop reasons that you can track by using the `show dropcounters capabilities` command. By default, there are two types of drop counters:
  - PORT\_INGRESS\_DROPS — Frames dropped from the ingress forwarding pipeline on the port.
  - PORT\_MIRROR\_SUPPORTED\_INGRESS\_DROPS — Dropped frames that are assigned to a mirroring session to the CPU or a specified port.

The reasons for each drop type are listed below the drop counter name. The number of dropped packets for each counter type is also displayed. For example:

```
sonic# show dropcounters capabilities
Counter Type Total

PORT_INGRESS_DROPS 3
PORT_MIRROR_SUPPORTED_INGRESS_DROPS 5

PORT_INGRESS_DROPS:
 ANY
 MPLS_MISS
 IP_HEADER_ERROR
 FDB_AND_BLACKHOLE_DISCARDS
 SMAC_EQUALS_DMAC
 ACL_ANY
 SIP_LINK_LOCAL
 DIP_LINK_LOCAL
 L3_EGRESS_LINK_DOWN
```

```
EXCEEDS_L3_MTU
PORT_MIRROR_SUPPORTED_INGRESS_DROPS:
 ANY
```

2. (Optional) Create a custom drop counter. Enter the counter name as a text string; 32 characters maximum. The counter name must start with an alphanumeric character.

```
sonic(config) # dropcounters counter-name
sonic(config-dropcounters-name) #
```

To delete a custom drop counter, enter the `no dropcounters counter-name` command.

3. Configure a custom drop counter:

- Add a drop reason to the new counter. Re-enter the command to configure additional reasons. To view the list of valid drop reasons, enter the `show dropcounters capabilities` command.

```
sonic(config-dropcounters-name) # add-reason drop-reason-name
```

To delete a drop reason, enter the `delete-reason drop-reason-name` command.

- (Optional) Enter a description of the new counter; 240 characters maximum.

```
sonic(config-dropcounters-name) # description text
```

- Assign the new counter to one of the default counter types: `PORT_INGRESS_DROPS` or `PORT_MIRROR_SUPPORTED_INGRESS_DROPS`. To view the list of valid counter types, enter the `show dropcounters capabilities` command.

```
sonic(config-dropcounters-name) # type counter-type
```

To delete the counter type assignment, enter the `no type counter-type` command.

- Add the new counter to a drop counter group. If the new counter records legitimate drops, add it to the default `RX_LEGIT` group. If the new counter records non-legitimate drops, you can add it to a custom drop group. Counters for the `RX_LEGIT` and custom drop groups are displayed in `show interface dropcounters` outputs. If you do not assign the new counter to a drop group, packet drops for the new counter are displayed next to the counter name in `show` outputs.

```
sonic(config-dropcounters-name) # group drop-group-name
```

To remove a drop group assignment, enter the `no group` command.

- (Optional) Assign a user-friendly alias to use in place of the new counter name; 24 characters maximum. If no alias is configured, you must enter the counter name to identify it.

```
sonic(config-dropcounters-name) # alias text
```

To remove a counter alias, enter the `no alias` command.

- (Optional) Add a new counter to an existing mirror session. To set up a mirror session, see [Port monitoring](#). Default counter types are included by default in mirror sessions.

```
sonic(config-dropcounters-name) # mirror-session session-name
```

 **NOTE:** When the mirror session is active, a drop counter is installed in the ASIC on the destination device. When the mirror session is inactive, the drop counter is removed from ASIC.

To remove a custom drop counter from a mirror session, enter the `no mirror-session` command.

4. Enable a custom drop counter. If you have not configured the mandatory parameters for a custom drop counter — `add-reason`, `type`, and `group` — an error message is displayed.

```
sonic(config-dropcounters-name) # enable
```

To disable drop counter recording, enter the `no enable` command.

## View drop counter configuration and statistics

### View current drop counter configuration

```
sonic# show dropcounters configuration [detail]

sonic# show dropcounters configuration

Counter Alias Group Type Mirror Reasons
----- ----- ----- -----
RX_LINK1 Link LEGIT PORT_INGRESS_DROPS SIP_LINK_LOCAL,
RX_IPHE IPHeader LEGIT PORT_INGRESS_DROPS IP_HEADER_ERROR,

sonic# show dropcounters configuration detail

Counter : RX_LINK1
Description : Link Local drops
Alias :
Group :
Type :
Mirror :
Reasons :
Status : Enabled

Counter : RX_IPHE
Description : IP header drops
Alias :
Group :
Type :
Mirror :
Reasons :
Status : Enabled
```

### View interface-level drop counters

```
sonic# show interface dropcounters [Ethslot/port]

sonic# show interface dropcounters

IFACE STATE RX_ERR RX_DROPS TX_ERR TX_DROPS RX_LINK1 RX_IPHE
----- ----- ----- ----- ----- ----- -----
Eth1/1 U 10 100 0 0 20 0
Eth1/2 U 0 1000 0 0 100 0
Eth1/3 U 100 10 0 0 0 0
```

- STATE — Interface state displays as UP (U) or DOWN (D).
- RX\_ERR — Frames with Cyclic Redundancy Check (CRC) or any other error that are dropped before entering the ingress forwarding pipeline.
- RX\_DROPS — Frames dropped in the ingress pipeline.
- TX\_ERR — Frames with a CRC error and any other error frames.
- TX\_DROPS — Frames dropped in the egress pipeline.
- RX\_LINK1 — Frames dropped with Session Initiation Protocol (SIP) link-local errors.
- RX\_IPHE — Frames dropped with IP header errors.

```
sonic# show interface dropcounters Eth1/1

IFACE STATE RX_ERR RX_DROPS TX_ERR TX_DROPS RX_LINK1 RX_IPHE
----- ----- ----- ----- ----- ----- -----
Eth1/1 U 10 100 0 0 20 0
```

### Clear drop counters thresholds

Use the `clear counters interface` command to delete all switch-level counters, all interface-level counters, or all drop counters on a specified interface.

```
sonic# clear counters interface {all | Ethslot/port}

sonic# clear counters interface all
Clear all interface drop counters [confirm y/N]: y

sonic# clear counters interface Eth1/1
Clear counters for Eth1/1 [confirm y/N]: n
```

## Example: Drop counter configuration

```
Create and install drop counter
SONiC(config)# dropcounters DEBUG_2
SONiC(config-dropcounters-DEBUG2)# description "More port ingress drops"
SONiC(config-dropcounters-DEBUG2)# group "BAD"
SONiC(config-dropcounters-DEBUG2)# alias "BAD_DROPS"
SONiC(config-dropcounters-DEBUG2)# add-reason ANY
SONiC(config-dropcounters-DEBUG2)# add-reason EXCEEDS_L2_MTU
SONiC(config-dropcounters-DEBUG2)# type PORT_INGRESS_DROPS
SONiC(config-dropcounters-DEBUG2)# mirror Session1
SONiC(config-dropcounters-DEBUG2)# enable

Delete drop counter
SONiC(config)# no dropcounters DEBUG_2

Delete drop reasons from counter
SONiC(config)# dropcounters DEBUG_2
SONiC(config-dropcounters-DEBUG2)# delete EXCEEDS_L2_MTU
```

## Buffer thresholds to detect congestion

**(i) NOTE:** Buffer threshold configuration is available only in the Cloud Standard, Cloud Premium, Enterprise Standard, and Enterprise Premium bundles. It is not available in the Edge Standard bundle.

To track and receive real-time notification of congestion events in the network, configure thresholds on the memory buffers of ingress and egress interfaces. When a buffer threshold is exceeded, a notification is generated and recorded in the `Counters_DB` database.

Buffer thresholds are supported on physical port interfaces, the switch-level shared buffer pool, and switch-level ingress and egress buffer pools. Buffer thresholds are not configured by default. Port-channel interfaces and breakout interfaces are not supported.

To configure a buffer threshold for ingress or egress traffic, use the `threshold` command. To enable the monitoring of ingress and egress buffers on port queues, use the watermark feature (see [Quality of Service](#)).

When a buffer threshold is exceeded, a breach entry is recorded which includes:

- The counter on which the threshold was breached.
- On port-level buffers, the port associated with the counter on which the breach occurred.
- The ingress priority group or egress queue on which the breach occurred
- Percentage of buffer usage at the time of the breach
- Total buffer usage (in bytes) when the breach occurred
- Time stamp

To view a list of threshold breach events, use the `show threshold breaches` command.

Threshold counters are maintained (in bytes) for port-level and switch-level buffer pools:

- Per-port counters for:
  - Ingress priority-group shared buffers
  - Ingress priority-group headroom buffers

- Egress queue buffers
- Ingress unicast and multicast shared buffers
- Global switch-level counters for:
  - Egress unicast and multicast buffers
  - Global switch-level memory buffer

Configure a buffer threshold as a percentage of buffer allocation. You can clear a configured threshold at any time. Configure a threshold on:

- One or more of the eight priority groups in the buffer of an ingress port.
- One or more of the unicast and multicast queues in the buffer of an egress port.

### Configure buffer thresholds

- Configure the threshold for priority-group traffic in the shared or headroom buffer on an ingress port interface. The *priority-group* is 0 to 7. The *threshold\_percentage* is 1 to 100. To reconfigure a shared or headroom buffer threshold and override a previously configured setting, re-enter the command.

```
sonic(conf-if-Eth) # threshold priority-group priority-group {headroom | shared}
threshold_percentage
```

- **shared** — Specifies the threshold for shared memory buffer usage for a priority group.
- **headroom** — Specifies the additional buffer limit that can be used when the shared buffer is exhausted, such as when flow control is enabled.

```
sonic(config) interface Eth1/2
sonic(conf-if-Eth1/2) # threshold priority-group 7 headroom 7
sonic(conf-if-Eth1/2) # threshold priority-group 1 shared 20
```

To remove the threshold configured for a shared or headroom priority-group buffer on an ingress port interface, enter the `no threshold priority-group priority-group {headroom | shared}` command.

- Configure the threshold for a unicast or multicast queue on an egress port interface. The *queue-index* specifies the queue number from 0 to 7. The *threshold\_percentage* is 1 to 100. Re-enter the command to configure additional unicast or multicast queue thresholds on the interface.

```
sonic(conf-if-Eth) # threshold queue queue-index {unicast | multicast}
threshold_percentage
```

```
sonic(config) interface Eth1/2
sonic(conf-if-Eth1/2) # threshold queue 5 multicast 5
sonic(conf-if-Eth1/2) # threshold queue 1 unicast 10
```

To remove the threshold configured for a unicast or multicast queue buffer on an egress port interface, enter the `no threshold queue queue-index {unicast | multicast}` command.

- Configure the threshold for the multicast queue buffer on the CPU interface. The *queue-index* specifies the queue number from 0 to 7. The *threshold\_percentage* is 1 to 100. Re-enter the command to configure additional multicast queue thresholds on the CPU.

```
sonic(conf-if-CPU) # threshold queue queue-index multicast threshold_percentage
```

```
sonic(config) # interface CPU
sonic(conf-if-CPU) # threshold queue 47 multicast 47
```

To remove the threshold configured for the multicast queue buffer on the CPU interface, enter the `no threshold queue queue-index multicast` command.

- Configure the threshold used for ingress and egress buffers in the switch-level buffer pool or a specified port-level buffer pool. The *threshold\_percentage* is 1 to 100. Buffer pools are used on ingress and egress interfaces for lossy or lossless traffic. To display a list of buffer-pool names, use the `show buffer pool` command.

```
sonic(config)# threshold buffer-pool buffer-pool-name {multicast | shared}
threshold_percentage
```

```
sonic(config)# interface Ethsport/port
sonic(conf-if-Eth)# threshold buffer-pool buffer-pool-name {shared | unicast}
threshold_percentage
```

To remove the threshold configured for ingress and egress buffers that are shared in the switch- or a port-level buffer pool, enter the `no threshold buffer-pool buffer-pool-name` command.

- Configure the threshold used for the global switch-level buffer pool. The *threshold\_percentage* is 1 to 100.

```
sonic(config)# threshold device threshold_percentage
```

To remove the threshold configured for the global switch-level buffer pool, enter the `no threshold device` command.

### **View buffer threshold configurations**

```
sonic# show threshold priority-group {headroom | shared}
sonic# show threshold queue {unicast | multicast | CPU}
sonic# show threshold buffer-pool
sonic# show threshold device
```

### **Example: Buffer threshold configurations**

```
sonic(config) interface Eth1/2
sonic(conf-if-Eth1/2)# threshold priority-group 7 headroom 7
sonic(conf-if-Eth1/2)# threshold priority-group 1 shared 20

sonic(config) interface Eth1/3
sonic(conf-if-Eth1/3)# threshold queue 5 multicast 5
sonic(conf-if-Eth1/3)# threshold queue 1 unicast 10

sonic(config) interface Ethernet CPU
sonic(conf-if-CPU)# threshold queue 47 multicast 47
```

| Port   | PG0 | PG1 | PG2 | PG3 | PG4 | PG5 | PG6 | PG7 |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| Eth1/1 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 7   |
| Eth1/2 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Eth1/3 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Eth1/4 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| ...    |     |     |     |     |     |     |     |     |

| Port   | PG0 | PG1 | PG2 | PG3 | PG4 | PG5 | PG6 | PG7 |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| Eth1/1 | 0   | 20  | 0   | 0   | 0   | 0   | 0   | 0   |
| Eth1/2 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Eth1/3 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Eth1/4 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| ...    |     |     |     |     |     |     |     |     |

| Port   | UC0 | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|
| Eth1/1 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

```

Eth1/2 0 10 0 0 0 0 0 0
Eth1/3 0 0 0 0 0 0 0 0
Eth1/4 0 0 0 0 0 0 0 0
...

```

```
sonic# show threshold queue CPU
```

```

Queue Percent

CPU:0 0
CPU:1 0
CPU:2 0
CPU:3 0
...
CPU:44 0
CPU:45 0
CPU:46 0
CPU:47 47
```

```
sonic# show threshold queue multicast
```

```

Port MC0 MC1 MC2 MC3 MC4 MC5 MC6 MC7

Eth1/1 0 0 0 0 0 0 0 0
Eth1/2 0 0 0 0 0 5 0 0
Eth1/3 0 0 0 0 0 0 0 0
Eth1/4 0 0 0 0 0 0 0 0
...

```

```
sonic# show threshold buffer-pool
```

```

pool_name type threshold

egress_lossless_pool shared 50
```

```
sonic# show threshold device
Threshold: 50
```

#### Example: View buffer threshold breaches

```
sonic# show threshold breaches

Event-id Buffer Type Port Index Breach Value(%) Time-stamp

2 priority-group shared Ethernet0 7 77 2020-04-14-11:35:20
3 queue unicast Ethernet0 5 45 2020-04-17-11:30:20
```

#### Clear buffer threshold breaches

Use the `clear threshold breach` command to delete the threshold breaches recorded on the switch. To delete a specified breach event, enter the `event-id` displayed in `show threshold breaches` output.

```
sonic# clear threshold breach {all | event-id}
```

```
sonic# clear threshold breach 100
```

```
sonic# clear threshold breach all
```

## Debug application for congestion and drops

**i** **NOTE:** The debug application is available only in the Enterprise Premium bundle. It is not available in the Cloud Standard, Cloud Premium, Enterprise Standard, and Edge Standard bundles.

To debug and resolve congestion and packet drops, use the debug application to collect HW and SW data from different sources on the switch. The aggregated data includes historical and real-time counters that are presented in a single, unified view.

### Usage notes

- By default, the periodic collection of data from HW and SW sources on the switch is disabled. When enabled, data is collected from all data sources every 15 seconds by default. You can reconfigure the default interval. The collected data is stored locally for future reference.
- The debug application holds data only for a configurable time period (by default, 90 minutes or 1.5 hours). You can clear the historical data that is collected on a per-tool basis.

### Configure data collection

Configure the time intervals for collecting and retaining HW and SW data on congestion events and packet drops. Enter the time interval at which data is collected (0 to 3600 seconds; default 15). 0 disables data collection.

```
admin@sonic:~$ bdbg config collection-interval seconds
```

Enter the maximum retention interval that historical data is held (0 to 5400 seconds; default 5400, 1.5 hours). After the configured time, all collected data is deleted. 0 indicates that no data is retained.

```
admin@sonic:~$ bdbg config max-retention-interval seconds
```

For example:

```
admin@sonic:~$ bdbg config collection-interval 100
admin@sonic:~$ bdbg config max-retention-interval 3600
```

To reset the data collection or retention intervals to their default values, enter:

```
admin@sonic:~$ bdbg clear collection-interval
```

```
admin@sonic:~$ bdbg clear max-retention-interval
```

### Configure congestion level for data buffers

- Configure the percentage of shared memory buffers that can be used (0 to 100; default 100). After this buffer percentage is reached, the buffer is considered to be under congestion and congestion events are recorded.

```
admin@sonic:~$ bdbg config congestion congestion-threshold buffer-percentage
```

For example:

```
admin@sonic:~$ bdbg config congestion congestion-threshold 40
```

To reset the percentage of all shared memory buffers, which can be used before buffer congestion events are recorded, to the default value, enter:

```
admin@sonic:~$ bdbg clear congestion congestion-threshold
```

### Start data collection

To start the data collection and monitoring of congestion and/or packet-drop events, enter the `bdbg start {all | congestion | drops}` command.

```
admin@sonic:~$ bdbg start {all | congestion | drops}
```

To stop data collection, enter the `bdbg stop {all | congestion | drops}` command.

```
admin@sonic:~$ bdbg stop {all | congestion | drops}
```

### Using a TAM collector

On supported Enterprise SONiC switches, you can use per-flow packet drop monitoring. To enable per-flow packet drop monitoring, configure a collector to receive the packets locally by using the `collector` command in Telemetry and Monitoring

(TAM) mode. Per-flow packet drops are then collected with other HW and SW data. Enter the following values with the command:

- Collector name: local
- Collector IP address: A locally configured IP addresses on the switch that is assigned to a routing interface
- Collector UDP port: 8250

For example:

```
sonic(config)# tam
sonic(conf-tam)# collector local ip 20.20.20.4 port 8250 protocol udp
```

### View data collection configuration

```
admin@sonic:~$ bdbg show
Supported tools : congestion, drops

Tuning Parameters

collection-interval : 5 sec
max-retention-interval : 3600 sec
```

### View congestion events

```
admin@sonic:~$ bdbg show congestion

Number of Congestion Events : 360
Switch Latency - Minimum : less than 1.2us (70%)
Switch Latency - Median : 10us - 20us (10%)
Switch Latency - Maximum : above 120us (1%)
Congestion Events last cleared at : 30th Jun 2021, 10:11AM
```

### View congestion events during current time interval

```
admin@sonic:~$ bdbg show congestion active

Active Monitoring interval : 30th Jun 2021, 10:11AM to 30th Jun 2021, 10:21AM
 Utilization %
Buffer Interface Direction SH HR UC MC Drops

device
queue 0 Ethernet64 Egress 38 0 0 0 0
PG 7 Ethernet46 Ingress 19 0 0 0 0
PG 7 Ethernet47 Ingress 19 0 0 0 0
ingress_lossless_pool Ethernet46 Ingress 19 0 0 0 0
ingress_lossless_pool Ethernet47 Ingress 19 0 0 0 0
egress_lossless_pool Ethernet64 Egress 0 0 4 0 0
egress_lossy_pool Ethernet64 Egress 4 0 4 0 0
egress_lossless_pool Ethernet64 Egress 38 0 0 0 0
ingress_lossless_pool Ethernet64 Ingress 100 0 0 0 0
```

Utilization columns display the percentages of buffer use for the following traffic types:

- SH — Shared buffer for unicast and multicast traffic
- HR — Headroom packets
- UC — Unicast packets
- MC — Multicast packets

**i | NOTE:** Not all traffic types are supported in each buffer — see the silicon documentation.

### View congestion sources

View the main congestion sources recorded since the congestion history was cleared. Historical records are deleted when they exceed the configured bdbg config max-retention-interval value. To limit the number of congestion sources displayed, enter a *num-sources* value (default 10).

```
admin@sonic:~$ bdbg show congestion top [limit num-sources]
```

```
admin@sonic:~$ bdbg show congestion top
```

| Number of Congestion Events       |            |           | :             | 360                    |    |       |       |
|-----------------------------------|------------|-----------|---------------|------------------------|----|-------|-------|
| Congestion Events last cleared at |            |           | :             | 30th Jun 2021, 10:11AM |    |       |       |
| Buffer                            | Interface  | Direction | Utilization % |                        |    |       |       |
|                                   |            |           | SH            | HR                     | UC | MC    | Drops |
| device                            |            |           | 38            | 0                      | 0  | 0     | 17    |
| PG 7                              | Ethernet47 | Ingress   | 19            | 0                      | 0  | 0     | 17    |
| queue 0                           | Ethernet64 | Egress    | 38            | 0                      | 0  | 21875 | 17    |

To view the congestion events recorded for a specified source since the congestion history was cleared, enter the name of the buffer or an Ethernet interface queue.

```
admin@sonic:~$ bdbg show congestion history {buffer buffer-name | interface Ethsport/port
queue queue-number} [limit num-events] [around DD:MM:YYYY:HH:MM:SS]
```

- buffer *buffer-name*** — Enter the name of a buffer to display the congestion events recorded for the buffer. The system administrator assigns buffer names during switch configuration, including the switch-level buffer name. To display buffer names, enter the show buffer pool command.
- interface *Ethsport/port* queue *queue-number*** — Enter the interface slot and port, and the queue number (0 - 7).
- limit *num-events*** — (Optional) Enter the number of congestion events to display (default 10).
- around *DD:MM:YYYY:HH:MM:SS*** — (Optional) Enter a time in *day:month:year:hour:minute:seconds* format to display the number of congestion events recorded since a specified time.

```
admin@sonic:~$ bdbg show congestion history interface Eth1/1 queue 8

Observation Point : Eth1/1 Queue 8
Number of Congestion Events : 3
Congestion Events last cleared at : 30th Jun 2021, 10:11AM

Id Timestamp Utilization Drops
--- -----
1 30th Jun 2021, 10:11AM 25% 0
2 30th Jun 2021, 10:21AM 25% 0
3 30th Jun 2021, 11:11AM 50% 0
```

```
admin@sonic:~$ bdbg show congestion history buffer pfc0
```

| Observation Point                 |                        |             | :     | pfc0 [Global Service Pool 0] |
|-----------------------------------|------------------------|-------------|-------|------------------------------|
| Number of Congestion Events       |                        |             | :     | 3                            |
| Congestion Events last cleared at |                        |             | :     | 30th Jun 2021, 10:11AM       |
| Id                                | Timestamp              | Utilization | Drops |                              |
|                                   |                        |             | ----- | -----                        |
| 1                                 | 30th Jun 2021, 10:11AM | 25%         | 0     |                              |
| 2                                 | 30th Jun 2021, 10:21AM | 25%         | 0     |                              |
| 3                                 | 30th Jun 2021, 11:11AM | 50%         | 0     |                              |

To delete all congestion events in the historical record, enter the bdbg clear congestion history command.

### View packet drops

```
admin@sonic:~$ bdbg show drops

Number of Dropping Flows : 3
Number of Active Drop Reasons : 2
Number of Active Drop Locations : 3
Number of Active Dropping CPU Queues : 3
Drop Events last cleared at : 30th Jun 2021, 10:11AM
```

To delete all packet-drop events in the historical record, enter the `bdbg clear drops history` command.

#### View packet drops during current time interval

```
admin@sonic:~$ bdbg show drops active {flows [detail drop-id] | reasons | locations | interface Ethslot/port}
```

- `flows` — Display the dropped flows in the data-plane (silicon), CPU queue, or kernel and the reason why the flow was dropped. The dropped flow is identified by the 5-tuple that identifies a TCP/IP connection: source and destination IP address, source and destination L4 port number and protocol (for example, 17 for UDP, 6 for TCP).
- `flows detail drop-id` — Display detailed packet-drop information for a specified `drop-id`.
- `reasons` — Display the packet-drop count and reasons why different flows have been dropped.
- `locations` — Display the packet-drop count in hardware (silicon) and software.
- `interface Ethslot/port` — Display the packet-drop count on a specified interface.

```
admin@sonic:~$ bdbg show drops active flows

Number of Dropping Flows : 3
Active Monitoring interval : 30th Jun 2021, 10:11AM to 30th Jun 2021, 10:21AM

drop-id src-ip dst-ip drop-reason drop-type Ingress Intf timestamp
----- ----- ----- ----- ----- ----- -----
2022 10.10.1.1 10.10.2.2 L3_DEST_MISS Active Ethernet24 2021-06-11 10:16AM
2023 10.10.1.2 10.10.2.2 L3_DEST_MISS Stopped Ethernet24 2021-06-11 10:17AM
2024 10.10.1.3 10.10.2.2 L3_DEST_MISS Active Ethernet24 2021-06-11 10:17AM
```

The `drop-type` column displays the type of flow drop event:

- `Active` — There are on-going packet drops in the specified flow.
- `Stopped` — A flow no longer experiences packet drops.

**(i) NOTE:** Active and Stopped drop events are not both available for all types of packet drops on all Enterprise SONiC switches. For example, some switches support only Active drop events.

```
admin@sonic:~$ bdbg show drops flows detail 2022

Number of Dropping Flows : 3
Active Monitoring interval : 30th Jun 2021, 10:11AM to 30th Jun 2021, 10:21AM

Flow Id : 2022
Src-Ip : 10.10.1.1
Dst-Ip : 10.10.2.2
Src-Port : 5656
Dst-Port : 80
protocol : 6
Src-Mac : 00:22:44:33:55:00
Dst-Mac : 00:22:44:33:55:01
Vlan-Id : 2043

Ingress Port : Ethernet24

Drop Reason : L3_DEST_MISS
Drop Event Type: Active
Drop Location : Ingress Pipeline
Drop Timestamp : 30th Jun 2021, 10:16AM
```

```
admin@sonic:~$ bdbg show drops active reasons

Number of Active Drop Reasons : 2
Active Monitoring interval : 30th Jun 2021, 10:11AM to 30th Jun 2021, 10:21AM

drop-reason count
----- -----
L3_DEST_MISS 24
UNKNOWN_VLAN 32
```

```
admin@sonic:~$ bdbg show drops active locations
```

```

Number of Active Drop Locations : 3
Active Monitoring interval : 30th Jun 2021, 10:11AM to 30th Jun 2021, 10:21AM

location count
----- -----
data-plane 3400
cpu-queue 32
kernel 45

```

```

admin@sonic:~$ bdbg show drops active interface cpu

Number of Active Dropping Queues : 3
Active Monitoring interval : 30th Jun 2021, 10:11AM to 30th Jun 2021, 10:21AM

location count
----- -----
Queue 0 [Unassigned] 3400
Queue 24 [BFD] 25
Queue 36 [MOD] 10

```

```

admin@sonic:~$ bdbg show drops active interface Eth1/1

Number of Active Dropping Queues : 1
Active Monitoring interval : 30th Jun 2021, 10:11AM to 30th Jun 2021, 10:21AM

location count
----- -----
Queue 0 5

```

#### **View packet drops since drop history was cleared**

To view the packet drops recorded since the drop history was cleared, enter the `bdbg show drops history` command. Historical records are deleted when they exceed the configured `bdbg config max-retention-interval` value and are not counted in packet drop statistics.

```

admin@sonic:~$ bdbg show drops history {flows | reasons | locations | interface Ethsport/
port}
[limit num-events] [around DD:MM:YYYY:HH:MM:SS]

```

- `flows` — Display the dropped flows in the data-plane (silicon), CPU queue, or kernel with the reason why the flow was dropped. The dropped flow is identified by a 5-tuple that identifies a TCP/IP connection: source and destination IP address, source and destination L4 port number and protocol (for example, 17 for UDP, 6 for TCP).
- `reasons` — Display the packet-drop count and reasons why different flows have been dropped.
- `locations` — Display the packet-drop count in hardware (silicon) and software.
- `interface Ethsport/port` — Display the packet-drop count on a specified interface.
- `limit num-events` — (Optional) Enter the number of drop events to display (default 10).
- `around DD:MM:YYYY:HH:MM:SS` — (Optional) Enter a time in `day:month:year:hour:minute:seconds` format to display the number of drop events recorded since a specified time.

```

admin@sonic:~$ bdbg show drops history flows

Number of Dropping Flows : 3
Drop Events last cleared at : 30th Jun 2021, 10:11AM

src-id dst-ip src-port dst-port protocol drop-reason location drop-type Ingress Intf timestamp
----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
10.10.1.1 10.10.2.2 5656 80 6 L3_DEST_MISS data-plane Active Ethernet24 2021-06-11 11:22AM
10.10.1.1 10.10.2.2 5656 80 6 UNKNOWN_VLAN cpu-queue Active Ethernet24 2021-06-11 11:20AM
10.10.1.1 10.10.2.2 5656 80 6 UNKNOWN_VLAN kernel Active Ethernet24 2021-06-11 11:20AM

```

```

admin@sonic:~$ bdbg show drops history reasons

Number of Active Drop Reasons : 2
Drop Events last cleared at : 30th Jun 2021, 10:11AM

drop-reason count
----- -----

```

```
L3_DEST_MISS 240
UNKNOWN_VLAN 320
```

```
admin@sonic:~$ bdbg show drops history locations
Number of Active Drop Locations : 3
Drop Events last cleared at : 30th Jun 2021, 10:11AM

location	count
data-plane	34000
cpu-queue	320
kernel	450


```

```
admin@sonic:~$ bdbg show drops history interface CPU
Number of Active Dropping Queues : 3
Drop Events last cleared at : 30th Jun 2021, 10:11AM

location	count
Queue 0 [Unassigned]	34000
Queue 24 [BFD]	250
Queue 36 [MOD]	100


```

```
admin@sonic:~$ bdbg show drops history interface Eth1/1
Number of Active Dropping Queues : 1
Drop Events last cleared at : 30th Jun 2021, 10:11AM

location	count
Queue 0	50


```

### Clear congestion history

```
admin@sonic:~$ bdbg clear congestion history
```

### Clear drop event history

```
admin@sonic:~$ bdbg clear drops history
```

### Clear congestion and drop event history

```
admin@sonic:~$ bdbg clear history
```

### Display debug tool settings

To view a dump of the current settings used by internal debugging tools, use the `bdbg dump all` command from the Linux shell. The tool-specific parameters that are displayed are useful to software developers for debugging purposes.

```
admin@sonic:~$ bdbg dump all

admin@sonic:~$ bdbg dump all
root@sonic:/usr/bin# bdbg dump all

Congestion Tool:

Status = Started
Remaining time to clear history = 0
Congestion tool local tables:
Device ID map(device name -- vid)

device -- oid:0x2100000000000000

Port Queue map(port:queue --(port, queue, type, vid))

CPU:0 -- (CPU, 0, MCAST, oid:0x15000000000053)
```

```

CPU:1 -- (CPU, 1, MCAST, oid:0x150000000000054)
CPU:10 -- (CPU, 10, MCAST, oid:0x15000000000005d)
CPU:11 -- (CPU, 11, MCAST, oid:0x15000000000005e)
CPU:12 -- (CPU, 12, MCAST, oid:0x15000000000005f)
CPU:13 -- (CPU, 13, MCAST, oid:0x150000000000060)
CPU:14 -- (CPU, 14, MCAST, oid:0x150000000000061)
CPU:15 -- (CPU, 15, MCAST, oid:0x150000000000062)
CPU:16 -- (CPU, 16, MCAST, oid:0x150000000000063)
CPU:17 -- (CPU, 17, MCAST, oid:0x150000000000064)
CPU:18 -- (CPU, 18, MCAST, oid:0x150000000000065)
CPU:19 -- (CPU, 19, MCAST, oid:0x150000000000066)
CPU:2 -- (CPU, 2, MCAST, oid:0x150000000000055)
CPU:20 -- (CPU, 20, MCAST, oid:0x150000000000067)
CPU:21 -- (CPU, 21, MCAST, oid:0x150000000000068)
CPU:22 -- (CPU, 22, MCAST, oid:0x150000000000069)
...
CPU:6 -- (CPU, 6, MCAST, oid:0x150000000000059)
CPU:7 -- (CPU, 7, MCAST, oid:0x15000000000005a)
CPU:8 -- (CPU, 8, MCAST, oid:0x15000000000005b)
CPU:9 -- (CPU, 9, MCAST, oid:0x15000000000005c)
Ethernet0:0 -- (Ethernet0, 0, UCAST, oid:0x150000000000355)
Ethernet0:1 -- (Ethernet0, 1, UCAST, oid:0x150000000000356)
Ethernet0:10 -- (Ethernet0, 10, MCAST, oid:0x150000000000361)
Ethernet0:11 -- (Ethernet0, 11, MCAST, oid:0x150000000000362)
Ethernet0:12 -- (Ethernet0, 12, MCAST, oid:0x150000000000363)
Ethernet0:13 -- (Ethernet0, 13, MCAST, oid:0x150000000000364)
...
Ethernet92 -- oid:0x10000000000009
Ethernet96 -- oid:0x10000000000032
PortChannel1 -- oid:0x2000000000b83

Port Queue map(port:queue --(port, queue, type, vid))

CPU:0 -- (CPU, 0, MCAST, oid:0x150000000000053)
CPU:1 -- (CPU, 1, MCAST, oid:0x150000000000054)
CPU:10 -- (CPU, 10, MCAST, oid:0x15000000000005d)
CPU:11 -- (CPU, 11, MCAST, oid:0x15000000000005e)
CPU:12 -- (CPU, 12, MCAST, oid:0x15000000000005f)
CPU:13 -- (CPU, 13, MCAST, oid:0x150000000000060)
CPU:14 -- (CPU, 14, MCAST, oid:0x150000000000061)
CPU:15 -- (CPU, 15, MCAST, oid:0x150000000000062)
CPU:16 -- (CPU, 16, MCAST, oid:0x150000000000063)
CPU:17 -- (CPU, 17, MCAST, oid:0x150000000000064)
CPU:18 -- (CPU, 18, MCAST, oid:0x150000000000065)
CPU:19 -- (CPU, 19, MCAST, oid:0x150000000000066)
CPU:2 -- (CPU, 2, MCAST, oid:0x150000000000055)
CPU:20 -- (CPU, 20, MCAST, oid:0x150000000000067)
CPU:21 -- (CPU, 21, MCAST, oid:0x150000000000068)
CPU:22 -- (CPU, 22, MCAST, oid:0x150000000000069)
...
CPU:6 -- (CPU, 6, MCAST, oid:0x150000000000059)
CPU:7 -- (CPU, 7, MCAST, oid:0x15000000000005a)
CPU:8 -- (CPU, 8, MCAST, oid:0x15000000000005b)
CPU:9 -- (CPU, 9, MCAST, oid:0x15000000000005c)
Ethernet0:0 -- (Ethernet0, 0, UCAST, oid:0x150000000000355)
Ethernet0:1 -- (Ethernet0, 1, UCAST, oid:0x150000000000356)
Ethernet0:10 -- (Ethernet0, 10, MCAST, oid:0x150000000000361)
Ethernet0:11 -- (Ethernet0, 11, MCAST, oid:0x150000000000362)
Ethernet0:12 -- (Ethernet0, 12, MCAST, oid:0x150000000000363)
Ethernet0:13 -- (Ethernet0, 13, MCAST, oid:0x150000000000364)
...
Ethernet96:7 -- (Ethernet96, 7, UCAST, oid:0x15000000000085c)
Ethernet96:8 -- (Ethernet96, 8, UCAST, oid:0x15000000000085d)
Ethernet96:9 -- (Ethernet96, 9, UCAST, oid:0x15000000000085e)

Debug Counter map(debug_counter_name -- (reasons, stat_name))

COPP proto map(coppname -- (proto, queue))

copp-system-arp -- (arp, 10)
copp-system-bfd -- (bfd, 20)
copp-system-bgp -- (bgp, 14)

```

```
copp-system-dhcp -- (dhcp, 9)
copp-system-iccp -- (iccp, 16)
copp-system-icmp -- (icmp, 8)
copp-system-igmp -- (igmp, 12)
copp-system-ip2me -- (ip2me, 7)
copp-system-lacp -- (lacp, 23)
copp-system-lldp -- (lldp, 18)
copp-system-mtu -- (mtu, 4)
copp-system-nat -- (nat, 5)
copp-system-ospf -- (ospf, 15)
copp-system-pim -- (pim, 13)
copp-system-sflow -- (sflow, 3)
copp-system-stp -- (stp, 21)
copp-system-subnet -- (subnet, 6)
copp-system-suppress -- (suppress, 11)
copp-system-udld -- (udld, 22)
copp-system-vrrp -- (vrrp, 17)
default -- (default, 0)
```

```
LOCAL DB is copied to /tmp/bdbg-sql
```

## Isolate SONiC switch from network

If a SONiC switch is dropping traffic and behaving abnormally, you may want to isolate the device from the network for troubleshooting purposes.

Before isolating a device from the network, use `show techsupport` first (see [Basic troubleshooting](#)). You can then shut down BGP sessions to neighbors with `config bgp shutdown`.

1. Generate troubleshooting information.

```
sonic# show techsupport [since date Time]
```

2. Shut down BGP session with a neighbor.

```
sonic(conf-router-bgp)# neighbor {ip-address | ipv6-address | interface interface-type}
sonic(conf-router-bgp-neighbor)# shutdown
```

## NAT troubleshooting

This information describes the available commands for network address translation (NAT). All NAT-related configuration is saved in the REDIS database (CONFIG\_DB).

When you have IP connectivity problems in a NAT environment, it is often difficult to determine the cause of the problem. Many times NAT is mistakenly blamed, when in reality there is an underlying problem.

### Debug NAT issues

- Check if NAT global mode is enabled to display all NAT configuration.

```
sonic# show nat config

Global Values

Admin Mode : enabled
Global Timeout : 600 secs
TCP Timeout : 86400 secs
UDP Timeout : 300 secs

Static Entries

Nat Type IP Protocol Global IP Global Port Local IP Local Port Twice-NAT Id
----- -----
snat all 112.0.0.2 --- 111.0.0.3 --- 1

Pool Entries
```

```

Pool Name Global IP Range Global Port Range
----- -----
nat1 2.0.0.5 10-200

NAT Bindings

Binding Name Pool Name Access-List Nat Type Twice-NAT Id
----- -----
bind1 nat1 snat ---

NAT Zones

Port Zone
----- -----
Eth1/1 1
Eth1/2 2

```

- Check if the ingress and egress L3 interfaces are configured in different NAT zones.

```

sonic# show nat config zones

Port Zone
----- -----
Eth1/1 1
Loopback0 1
Vlan5 0
PortChannel12 2

```

- Check if the NAT miss packets are reaching the Linux stack to monitor the ingress L3 interface.

```
admin@sonic:~$ tcpdump
```

- Check if the NAT translation entries are learned by the Linux kernel. Use conntrack -j -L and iptables -t nat -nv -L.
- Check if the NAT translation entry is created, then view the statistics per NAT entry.

```
sonic# show nat translations
```

| Protocol | Source        | Destination     | Translated Source | Translated Destination |
|----------|---------------|-----------------|-------------------|------------------------|
| all      | 10.0.0.1      | ---             | 65.55.42.2        | ---                    |
| all      | ---           | 65.55.42.2      | ---               | 10.0.0.1               |
| all      | 10.0.0.2      | ---             | 65.55.42.3        | ---                    |
| all      | ---           | 65.55.42.3      | ---               | 10.0.0.2               |
| tcp      | 20.0.0.1:4500 | ---             | 65.55.42.1:2000   | ---                    |
| tcp      | ---           | 65.55.42.1:2000 | ---               | 20.0.0.1:4500          |
| udp      | 20.0.0.1:4000 | ---             | 65.55.42.1:1030   | ---                    |
| udp      | ---           | 65.55.42.1:1030 | ---               | 20.0.0.1:4000          |
| tcp      | 20.0.0.1:6000 | ---             | 65.55.42.1:1024   | ---                    |
| tcp      | ---           | 65.55.42.1:1024 | ---               | 20.0.0.1:6000          |
| tcp      | 20.0.0.1:5000 | 65.55.42.1:2000 | 65.55.42.1:1025   | 20.0.0.1:4500          |
| tcp      | 20.0.0.1:4500 | 65.55.42.1:1025 | 65.55.42.1:2000   | 20.0.0.1:5000          |

```
sonic# show nat statistics
```

| Protocol | Source              | Destination       | Packets | Bytes |
|----------|---------------------|-------------------|---------|-------|
| all      | 100.100.100.100     | 200.200.200.5     | 15      | 12785 |
| all      | 17.17.17.17         | 15.15.15.15       | 10      | 12754 |
| all      | 12.12.12.14         | ---               | 0       | 0     |
| all      | ---                 | 138.76.28.1       | 12      | 12500 |
| tcp      | 12.12.15.15:1200    | ---               | 0       | 0     |
| tcp      | ---                 | 138.76.29.2:250   | 8       | 85120 |
| tcp      | 100.100.101.101:251 | 200.200.201.6:276 | 21      | 21654 |
| tcp      | 17.17.18.18:1251    | 15.15.16.16:1201  | 18      | 21765 |

- Check if the NAT translation entries are installed in the switching ASIC. Use bcmcmd 13 nat\_ ingress show and 13 nat\_egress show Broadcom shell commands.

- Check if the packet counters for the installed NAT entries are incrementing to monitor the packet and byte counter per entry.

```
sonic# show nat statistics

Protocol Source Destination Packets Bytes
----- -----
all 100.100.100.100 200.200.200.5 15 12785
all 17.17.17.17 15.15.15.15 10 12754
all 12.12.12.14 --- 0 0
all --- 138.76.28.1 12 12500
tcp 12.12.15.15:1200 --- 0 0
tcp --- 138.76.29.2:250 8 85120
tcp 100.100.101.101:251 200.200.201.6:276 21 21654
tcp 17.17.18.18:1251 15.15.16.16:1201 18 21765
```

- Verify that the translated source IP address for the outbound traffic belongs to one of the L3 interfaces
- Check if the translated destination IP address for the inbound traffic is reachable using one of the L3 interfaces

## NAT clear commands

- Clear all NAT statistics

```
sonic# clear nat statistics
NAT statistics are cleared.
```

- Clear all dynamic NAT translations

```
sonic# clear nat translations
Dynamic NAT entries are cleared.
```

## Kernel dump

To debug a kernel crash, you can generate core files with a dump of the crash. You can analyze the line of code that caused the crash, then view the kernel logs of the events that led to the crash.

The dump files are in `show techsupport` output. Using the `show techsupport .tar` output file, you can export the kernel core files to a remote server for offline analysis (see *Diagnostic tools* in [Basic troubleshooting](#)). Use Debian and Linux kernel-provided native tools to view kernel cores.

When you configure the kdump feature, you can change the default and specify the amount of memory to reserve for the kernel core file. Be sure to leave enough available memory for protocol applications to operate. You can also configure the maximum number and size of the core files.

### Configure and enable kdump

- Enable kdump (disabled by default).

```
sonic(config) # kdump enable
KDUMP configuration has been updated in the startup configuration
Kdump configuration changes will be applied after the system reboots
```

To disable kdump, use the following command and reboot the switch.

```
sonic(config) # no kdump enable
KDUMP configuration has been updated in the startup configuration
ALERT! A system reboot is highly recommended.
Kdump configuration changes will be applied after the system reboots
```

- (Optional) Configure the amount of memory reserved for the kernel crash dump files. Specify the reserved memory in MB in the format `M`. The `M` parameter is mandatory. The amount of memory that is reserved by default for kernel crash dump files depends on the RAM size.

- 0 M to 2 GB RAM reserves 256M

- 2 GB to 4 GB reserves 320M
- 4 GB to 8 GB reserves 384M
- 8 GB reserves 448M

```
sonic(config) # kdump memory 512M
KDump configuration has been updated in the startup configuration
kdump updated memory will be only operational after the system reboots
```

To reset the amount of memory that is reserved for kernel core dump files to the default value, use no `kdump memory`, then reboot the switch.

3. (Optional) Configure the maximum number of kernel core dump files (1 to 9; default 3) that can be stored locally. If a new kernel core dump is generated that exceeds the configured number of files, the oldest stored file is deleted. Save the configuration change.

```
sonic(config) # kdump num-dump 5
```

To reset the maximum number of kernel core files that are stored locally to the default value, use no `kdump num_dumps`, then reboot the switch.

4. Verify the kdump configuration.

```
sonic# show kdump status
Kdump Administrative Mode: Enabled
Kdump Operational State: Ready after reboot
Memory Reserved: 512M
Memory Allocated: 512M
Maximum number of Kernel Core files Stored: 5
No kernel core dump files
```

```
sonic# show kdump memory
Memory Reserved: 512M
```

```
sonic# show kdump num-dumps
Maximum number of Kernel Core files Stored: 4
```

5. Reboot the switch to apply the kdump configuration settings.

```
sonic# reboot
```

### **View kernel core dump file after crash**

```
sonic# show kdump status
Kdump Administrative Mode: Enabled
Kdump Operational State: Ready
Memory Reserved: 512M
Maximum number of Kernel Core files Stored: 4
Record Key Filename

1 202003170138 /var/crash/202003170138/dmesg.202003170138
 /var/crash/202003170138/kdump.202003170138
```

```
sonic# show kdump files
Record Key Filename

1 202003170138 /var/crash/202003170138/dmesg.202003170138
 /var/crash/202003170138/kdump.202003170138
```

### **View kernel core file**

To display kernel core dump log from a locally stored file, use the `show kdump log record [lines]` command. Enter the record number of the kernel crash log displays in `show kdump files` output. The mandatory parameter is the number of the kernel core dump files which are stored locally. The optional parameter is the number of lines displayed (default 20).

```
sonic# show kdump log 1 5
File: /var/crash/202002101809/dmesg.202002101809
[326785.222049] [<fffffffffffa0c0484e>] ? entry_SYSCALL_64_after_swaps
+0x58/0xc6
[326785.229926] Code: 41 5c 41 5d 41 5e 41 5f e9 6c 2f cf ff 66 2e 0f 1f
84 00 00 00 00 66 90 0f 1f 44 00 00 c7 05 29 28 a8 00 01 00 00 00 0f
ae f8 <c6> 04 25 00 00 00 01 c3 0f 1f 44 00 00 0f 1f 44 00 00 53 8d
[326785.251451] RIP [<fffffffffffa0a2a562>] sysrq_handle_crash+0x12/0x20
[326785.258463] RSP <fffffafd2c6523e78>
[326785.262453] CR2: 0000000000000000
...
```

## Check memory usage

Applications can over-use or leak memory, resulting in an out-of-memory condition that impacts system performance. A memory leak occurs when system memory is unnecessarily allocated to a process. To monitor memory usage in system processes, use the memory histogram to collect memory data at regular intervals. View the memory usage in processes, dockers, and the system when you need to troubleshoot a memory-related issue.

The memory histogram is enabled by default to record memory usage at the process, docker, and system levels. The default settings are:

- Sampling interval — 3 minutes
- Data duration in histogram view — 30 days

### View memory usage

To view memory usage, enter the `show histogram memory` command. By default, 30 days of memory usage statistics are displayed. Specify `process`, `docker`, or `system` to limit the memory usage display.

```
sonic# show histogram memory [process | docker | system] [stime datetime] [etime
datetime] [filter string] [analyze leak]
```

- `{stime | etime} datetime` — (Optional) Enter a start time and end time for memory data collection in ISO format. Enclose the `datetime` in single quotes. For detailed information about how to enter ISO dates and times, see [Data input formats](#) and [Examples of datetime](#). Some `datetime` examples are:

```
'July 23' '2 days ago' 'now' '3 hours ago' '10 minutes ago'
'yesterday' 'today' 'july 1 3:00:00' 'Aug 01 06:43:40'
'1 july 2021' '2 am' '7/24'
```

- If you do not enter `stime` and `etime` `datetime` values, the last 30 days of memory usage are displayed by default.
- If you enter `stime` and `etime` `datetime` values, the date difference cannot be more than 30 days; the time difference cannot be less than 3 minutes.
- If you enter only an `stime` `datetime` value, the `etime` default is 'now'.
- If you enter only an `etime` `datetime` value, the `stime` default is 'now'-'30 days'.
- `filter string` — (Optional) To filter the memory data displayed in process, docker, or system memory output, enter a string to identify a text pattern or exact name.
- `analyze leak` — (Optional) Display memory leak data.

### Memory usage output

**i|NOTE:** If no memory data is available for a specified time range, an empty data display is returned.

- `Current` — Displays current memory use for a process, docker, or the system.
- `High/Low` — Displays the highest and lowest memory usage during the specified time range.
- `Process List` — Displays the process name and process ID or docker name and container ID. For system-level memory usage, the amounts of total, used, free, buffer, cached, and available memory are displayed.

### Example: View memory usage

To display the last 30 days of memory usage data for all processes:

```
sonic# show histogram memory process

Start Time : 2021-09-11 10:07:03
End Time : 2021-10-11 10:07:03
Current Time : 2021-10-11 10:07:03

Days Days
[11-14] [14-17] [17-20] [20-23] [23-26] [26-29] [29-02] [02-05] [05-08] [08-11] [11-11]
Sep/11 Sep/14 Sep/17 Sep/20 Sep/23 Sep/26 Sep/29 Oct/2 Oct/5 Oct/8 Oct/11 Current High/Low Process List
----- -----
- - - - - - - - - - - -
- - - - - - - - - - - -
- - - - - - - - - - - -
----- -----
504.8M 504.9M 504.8M/504.8M syncd(6778)
20.3M 20.3M 20.3M/20.3M python3(23259)
19.4M 19.4M 19.4M/19.4M sysmonitor.py(4347)
```

## View memory usage for processes

### Examples

To display the last 30 days of memory usage data for a specified process:

```
sonic# show histogram memory process filter bgp

Start Time : 2021-07-02 13:06:35
End Time : 2021-08-01 13:06:35
Current Time : 2021-08-01 13:06:35

Days Days
[02-05] [05-08] [08-11] [11-14] [14-17] [17-20] [20-23] [23-26] [26-29] [29-01] [01-01]
Jul1/2 Jul1/5 Jul1/8 Jul1/11 Jul1/14 Jul1/17 Jul1/20 Jul1/23 Jul1/26 Jul1/29 Aug/1 Current High/Low Process List
----- -----
- - - - - - - - - - - -
- - - - - - - - - - - -
----- -----
19.2M 19.2M 19.2M/19.2M bgpd(7073)
3.0M 3.0M 3.0M/3.0M bgp.sh(4181)
```

To display the memory usage data for a process during a specified time period — days, hours, or minutes:

```
sonic# show histogram memory process stime "5 days ago" etime "now" filter bgp

Start Time : 2021-07-27 13:10:42
End Time : 2021-08-01 13:10:42
Current Time : 2021-08-01 13:10:42

Days Days Days Days Days Days
[27-28] [28-29] [29-30] [30-31] [31-01] [01-01]
Jul1/27 Jul1/28 Jul1/29 Jul1/30 Jul1/31 Aug/1 Current High/Low Process List
----- -----
- - - - - - 19.2M 19.2M 19.2M/19.2M bgpd(7073)
- - - - - - 3.0M 3.0M 3.0M/3.0M bgp.sh(4181)
```

```
sonic# show histogram memory process stime "12 hours ago" etime "now" filter bgp

Start Time : 2021-08-01 01:12:52
End Time : 2021-08-01 13:12:52
Current Time : 2021-08-01 13:12:52

Hours Hours Hours Hours Hours Hours Hours
[01-03] [03-05] [05-07] [07-09] [09-11] [11-13] [13-15]
Aug/1 Aug/1 Aug/1 Aug/1 Aug/1 Aug/1 Aug/1 Current High/Low Process List
----- -----
- - - - - - 18.4M 18.6M 18.4M/18.4M bgpd(7073)
- - - - - - 3.2M 3.2M 3.2M/3.2M bgp.sh(4184)
- - - - - - 3.1M 3.1M 3.2M/3.2M bgp.sh(4181)
```

```
sonic# show histogram memory process stime "15 minutes ago" etime "now" filter bgp

Start Time : 2021-08-01 12:58:35
End Time : 2021-08-01 13:13:35
Current Time : 2021-08-01 13:13:35

Minutes Minutes Minutes Minutes Minutes Minutes Minutes
[58-00] [00-02] [02-04] [04-06] [06-08] [08-10] [10-12] [12-13]
12PM 01PM 01PM 01PM 01PM 01PM 01PM 01PM Current High/Low Process List
----- -----
- 18.6M - 18.6M 18.6M - 18.6M 18.6M 18.6M/18.6M bgpd(7073)
- 3.2M - 3.2M 3.2M - 3.2M 3.2M 3.2M/3.2M bgp.sh(4184)
- - - - - - 3.1M 3.1M 3.2M 3.2M/3.2M bgp.sh(4181)
```

### View possible memory leaks in processes

Use the memory histogram if you suspect a memory leak in a process, such as in the following output. In this example, the mem\_leak process starts from 752.0K memory use, increases to 5.6M, and then decreases to 1.5M.

```
sonic# show histogram memory process filter mem_leak stime "20 minutes ago"

Start Time : 2021-09-20 04:28:19
End Time : 2021-09-20 04:48:19
Current Time : 2021-09-20 04:48:19

Minutes Minutes Minutes Minutes Minutes Minutes Minutes
[28-31] [31-34] [34-37] [37-40] [40-43] [43-46] [46-48]
04AM 04AM 04AM 04AM 04AM 04AM Current High/Low Process List
----- ----- ----- ----- ----- ----- -----
- 752.0K 2.3M 3.3M 1.3M 5.6M 1.5M 1.5M 5.6M/752.0K mem_leak(20951)
```

To view memory leak data, enter the analyze leak parameters. The Diff column displays the total difference in memory data collected between the start and end times for a process. If the Diff value is high, it may indicate a memory leak. In the following output for the mem\_leak process, the Diff value of 744K shows a positive increase in memory use from base 752.0K at 4:31-34AM.

```
sonic# show histogram memory process filter mem_leak stime "20 minutes ago" analyze leak

Start Time : 2021-09-20 04:28:26
End Time : 2021-09-20 04:48:26
Current Time : 2021-09-20 04:48:26

Minutes Minutes Minutes Minutes Minutes Minutes Minutes
[28-31] [31-34] [34-37] [37-40] [40-43] [43-46] [46-48]
10AM 10AM 10AM 10AM 10AM 10AM Diff Process Leak
----- ----- ----- ----- ----- ----- -----
- 0B 1.6M 1.0M -2.0M 4.2M -4.1M 744.0K mem_leak(20951)
```

## View memory usage for dockers

### Examples

To display the last 30 days of memory usage data for all dockers:

```
sonic# show histogram memory docker

Start Time : 2021-07-02 13:17:34
End Time : 2021-08-01 13:17:34
Current Time : 2021-08-01 13:17:34

Days Days Days Days Days Days Days Days Days Days
[02-05] [05-08] [08-11] [11-14] [14-17] [17-20] [20-23] [23-26] [26-29] [29-01] [01-01]
Jul1/2 Jul5/ Jul8/ Jul11/ Jul14/ Jul17/ Jul20/ Jul23/ Jul26/ Jul29/ Aug1/ Current High/Low Docker List
----- ----- ----- ----- ----- ----- ----- ----- ----- -----
- - - - - - - - - - 22.7M 22.5M 23.2M 22.7M/22.5M radv(bfef843)
- - - - - - - - - - 31.9M 31.9M 32.7M 31.9M/31.9M sflow(6d4aeb2)
- - - - - - - - - - 24.9M 23.5M 45.5M 24.9M/23.5M telemetry
(a639856)
```

To display the last 30 days of memory usage data for a specified docker:

```
sonic# show histogram memory docker filter vrrp

Start Time : 2021-07-02 13:19:00
End Time : 2021-08-01 13:19:00
Current Time : 2021-08-01 13:19:00

Days Days Days Days Days Days Days Days Days Days
[02-05] [05-08] [08-11] [11-14] [14-17] [17-20] [20-23] [23-26] [26-29] [29-01] [01-01]
Jul1/2 Jul5/ Jul8/ Jul11/ Jul14/ Jul17/ Jul20/ Jul23/ Jul26/ Jul29/ Aug1/ Current High/Low Docker List
----- ----- ----- ----- ----- ----- ----- ----- ----- -----
- - - - - - - - - - 24.0M 24.1M 24.9M 24.1M/24.0M vrrp(5008479)
```

To display the memory usage data for dockers during a specified time period — days, hours, or minutes:

```
sonic# show histogram memory docker stime "2 days ago" etime "now"

Start Time : 2021-07-27 13:21:49
End Time : 2021-08-01 13:21:49
Current Time : 2021-08-01 13:21:49

Days Days Days
[30-31] [31-01] [01-01]
Jul30 Jul31 Aug/1 Current High/Low Docker List
----- ----- ----- ----- -----
```

```

- 22.8M 22.5M 23.3M 22.8M/22.5M radv(bfef843)
- 32.0M 31.9M 32.8M 32.0M/31.9M sflow(6d4aeb2)

```

```
sonic# show histogram memory docker stime "23 hours ago" etime "now" filter "swss"
```

```

Start Time : 2021-07-31 14:26:46
End Time : 2021-08-01 13:26:46
Current Time : 2021-08-01 13:26:46

```

| Hours [14-17] | Hours [17-20] | Hours [20-23] | Hours [23-02] | Hours [02-05] | Hours [05-08] | Hours [08-11] | Hours [11-13] | Current | High/Low | Docker List                     |
|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------|----------|---------------------------------|
| Jul/31        | Jul/31        | Jul/31        | Jul/31        | Aug/1         | Aug/1         | Aug/1         | Aug/1         | 49.4M   | 46.4M    | 52.3M 49.4M/44.7M swss(6fd8f6d) |
| 44.7M         | 44.8M         | 44.8M         | 44.7M         | 44.7M         | 44.7M         | 49.4M         | 44.7M         |         |          |                                 |

```
sonic# show histogram memory docker stime "23 minutes ago" etime "now" filter "swss"
```

```

Start Time : 2021-08-01 13:04:28
End Time : 2021-08-01 13:27:28
Current Time : 2021-08-01 13:27:28

```

| Minutes [04-07] | Minutes [07-10] | Minutes [10-13] | Minutes [13-16] | Minutes [16-19] | Minutes [19-22] | Minutes [22-25] | Minutes [25-27] | Current | High/Low | Docker List                     |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---------|----------|---------------------------------|
| 01PM            | 52.4M   | 52.3M    | 52.5M 52.4M/52.3M swss(6fd8f6d) |
| -               | -               | -               | -               | -               | -               | -               | -               |         |          |                                 |

## View possible memory leaks in dockers

Use the memory histogram if you suspect a memory leak in a docker, such as in the following output. In this example, the stp docker starts from 25.5M memory use, increases slightly to 26.2M, and then suddenly increases to 1.0G.

```
sonic# show histogram memory docker filter stp stime "20 minutes ago"
```

```

Start Time : 2021-10-11 10:25:30
End Time : 2021-10-11 10:45:30
Current Time : 2021-10-11 10:45:30

```

| Minutes [25-28] | Minutes [28-31] | Minutes [31-34] | Minutes [34-37] | Minutes [37-40] | Minutes [40-43] | Minutes [43-45] | Current | High/Low | Docker List                  |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---------|----------|------------------------------|
| 10AM            | 1.0G    | 1.0G     | 1.0G 1.0G/25.5M stp(1798e37) |
| -               | 25.5M           | 25.5M           | 25.5M           | 26.2M           | 26.2M           | 1.0G            |         |          |                              |

To view memory leak data, enter the analyze leak parameters. The Diff column displays the total difference in memory data collected between the start and end times for a docker. If the Diff value is high, it may indicate a memory leak. In the following output for the stp docker, the Diff value of 1006.7M shows a high increase in memory use from base 25.5M at 10:28-31AM.

```
sonic# show histogram memory process filter mem_leak stime "20 minutes ago" analyze leak
```

```

Start Time : 2021-10-11 10:25:51
End Time : 2021-10-11 10:45:51
Current Time : 2021-10-11 10:45:51

```

| Minutes [25-28] | Minutes [28-31] | Minutes [31-34] | Minutes [34-37] | Minutes [37-40] | Minutes [40-43] | Minutes [43-45] | Diff    | Docker Leak  |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---------|--------------|
| 10AM            | 1006.7M | stp(1798e37) |
| 0B              | 71.7K           | 0B              | 624.6K          | -471.0K         | 1006.5M         | -               |         |              |

## View system memory usage

### Examples

To display the last 30 days of system memory usage:

```
sonic# show histogram memory system
```

```

Start Time : 2021-09-11 11:06:56
End Time : 2021-10-11 11:06:56
Current Time : 2021-10-11 11:06:56

```

| Days [11-14] | Days [14-17] | Days [17-20] | Days [20-23] | Days [23-26] | Days [26-29] | Days [29-02] | Days [02-05] | Days [05-08] | Days [08-11] | Days [11-11] | Current | High/Low | System List |                                     |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------|----------|-------------|-------------------------------------|
| Sep/11       | Sep/14       | Sep/17       | Sep/20       | Sep/23       | Sep/26       | Sep/29       | Oct/2        | Oct/5        | Oct/8        | Oct/11       | Oct/11  | 15.1G    | 15.1G       | 15.3G 15.1G total                   |
| -            | -            | -            | -            | -            | -            | -            | -            | -            | -            | -            | -       | 2.5G     | 2.3G        | 2.5G/2.3G used                      |
| -            | -            | -            | -            | -            | -            | -            | -            | -            | -            | -            | -       | 11.0G    | 10.9G       | 11.0G/10.9G free                    |
| -            | -            | -            | -            | -            | -            | -            | -            | -            | -            | -            | -       | 251.2M   | 330.5M      | 318.1M 174.3M 330.5M/251.2M buffers |

```
- - - - - - - - - - - - 1.5G 1.6G 1.6G 1.3G 1.6G/1.5G cached
- - - - - - - - - - - - 12.4G 12.5G 12.7G 13.1G 12.7G/12.4G available
```

To display the last 30 days of system memory usage using a text filter:

```
sonic# show histogram memory system filter available

Start Time : 2021-09-11 11:08:18
End Time : 2021-10-11 11:08:18
Current Time : 2021-10-11 11:08:18

Days Days
[11-14] [14-17] [17-20] [20-23] [23-26] [26-29] [29-02] [02-05] [05-08] [08-11] [11-14]
Sep/11 Sep/14 Sep/17 Sep/20 Sep/23 Sep/26 Sep/29 Oct/2 Oct/5 Oct/8 Oct/11 Current High/Low System List
----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----
- - - - - - - - - - - - 12.4G 12.5G 12.7G 13.1G 12.7G/12.4G available
```

To display system memory usage during a specified time period days, hours, or minutes:

```
sonic# show histogram memory system stime "5 days ago" etime "now"

Start Time : 2021-10-06 11:08:46
End Time : 2021-10-11 11:08:46
Current Time : 2021-10-11 11:08:46

Days Days Days Days Days Days
[06-07] [07-08] [08-09] [09-10] [10-11] [11-11]
Oct/6 Oct/7 Oct/8 Oct/9 Oct/10 Oct/11 Current High/Low System List
----- ----- ----- ----- ----- -----
15.1G 15.1G 15.1G 15.1G 15.1G 15.3G 15.6G 15.3G/15.1G total
2.5G 2.4G 2.4G 2.3G 2.4G 2.3G 2.2G 2.5G/2.3G used
10.9G 11.0G 11.0G 10.9G 10.7G 11.0G 11.9G 11.0G/10.7G free
249.6M 251.9M 268.7M 344.2M 378.6M 318.1M 174.4M 378.6M/249.6M buffers
1.5G 1.5G 1.5G 1.6G 1.7G 1.6G 1.3G 1.7G/1.5G cached
12.3G 12.4G 12.5G 12.6G 12.5G 12.7G 13.0G 12.7G/12.3G available
```

```
sonic# show histogram memory system stime "20 hours ago" etime "now"

Start Time : 2021-10-10 15:09:21
End Time : 2021-10-11 11:09:21
Current Time : 2021-10-11 11:09:21

Hours Hours Hours Hours Hours Hours Hours Hours Hours Hours
15-17] [17-19] [19-21] [21-23] [23-01] [01-03] [03-05] [05-07] [07-09] [09-11] [11-11]
Oct/10 Oct/10 Oct/10 Oct/10 Oct/11 Oct/11 Oct/11 Oct/11 Oct/11 Oct/11 Oct/11 Current High/Low System List
----- ----- ----- ----- ----- -----
15.1G 15.1G 15.1G 15.1G 15.1G 15.1G 15.2G 15.6G 15.4G 15.6G 15.6G 15.6G/15.1G total
2.4G 2.4G 2.4G 2.4G 2.4G 2.4G 2.4G 2.2G 2.0G 2.5G 2.3G 2.5G/2.0G used
10.6G 10.6G 10.6G 10.6G 10.6G 10.6G 10.7G 11.9G 11.5G 11.6G 11.9G 11.9G/10.6G free
381.1M 381.9M 382.5M 383.1M 383.5M 384.0M 384.4M 369.3M 204.8M 225.8M 173.8M 174.5M 384.4M/173.8M buffers
1.8G 1.8G 1.8G 1.8G 1.8G 1.7G 1.7G 1.7G 1.3G 1.7G 1.3G 1.8G/1.3G cached
12.5G 12.5G 12.5G 12.4G 12.4G 12.4G 12.5G 13.0G 13.1G 12.7G 13.0G 13.1G/12.4G available
```

```
sonic# show histogram memory system stime "100 minutes ago" filter used

Start Time : 2021-10-11 09:30:00
End Time : 2021-10-11 11:10:00
Current Time : 2021-10-11 11:10:00

Minutes Minutes
[30-40] [40-50] [50-00] [00-10] [10-20] [20-30] [30-40] [40-50] [50-00] [00-10] [10-10]
09AM 09AM 10AM 10AM 10AM 10AM 10AM 10AM 11AM 11AM Current High/Low System List
----- ----- ----- ----- ----- -----
15.6G 15.6G 15.6G 15.1G 15.1G 15.4G 15.6G 15.6G 15.6G 15.6G 15.6G/15.1G total
1.6G 1013.2M 1.0G 2.1G 2.2G 2.2G 2.2G 2.3G 2.5G 2.5G 2.2G 2.5G/1013.2M used
12.4G 12.1G 7.7G 11.4G 11.3G 11.3G 11.7G 11.8G 11.6G 11.7G 11.9G 12.4G/7.7G free
223.0M 248.0M 319.9M 247.3M 249.8M 250.7M 195.1M 170.1M 172.7M 174.0M 174.6M 319.9M/170.1M buffers
1.3G 2.3G 6.5G 1.4G 1.4G 1.4G 1.3G 1.3G 1.3G 1.3G 1.3G 6.5G/1.3G cached
13.7G 14.3G 14.3G 12.7G 12.6G 12.7G 12.9G 13.0G 12.7G 12.8G 13.0G 14.3G/12.6G available
```

## View possible memory leaks in the system

Use the memory histogram if you suspect a memory leak in the system, such as in the following output. In this example, the system starts from base 2.2G memory use and increases to 5.2G.

```
sonic# show histogram memory system stime "100 minutes ago" filter used

Start Time : 2021-10-11 11:06:31
End Time : 2021-10-11 11:16:31
Current Time : 2021-10-11 11:16:31

Minutes Minutes
[06-07] [07-08] [08-09] [09-10] [10-11] [11-12] [12-13] [13-14] [14-15] [15-16] [16-16]
11AM Current High/Low System List
----- ----- ----- ----- -----
2.2G - - 2.2G - - 5.2G - - 5.2G - 5.2G 5.2G/2.2G used
```

To view memory leak data, enter the `analyze leak` parameters. The Diff column displays the total difference in memory data collected between the start and end times for the system. If the Diff value is high, it may indicate a memory leak. In the following output for the system, the Diff value of 3G shows an increase in used memory from 2.2G at 11:06-07AM.

```
sonic# show histogram memory system stime "10 minutes ago" filter used analyze leak
Start Time : 2021-10-11 11:06:41
End Time : 2021-10-11 11:16:41
Current Time : 2021-10-11 11:16:41

Minutes Minutes Minutes Minutes Minutes Minutes Minutes Minutes Minutes Minutes
[06-07] [07-08] [08-09] [09-10] [10-11] [11-12] [12-13] [13-14] [14-15] [15-16] [16-17]
11AM Diff System Leak
----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -
- - - 0.B - - 2.9G - - 8.9M - 3.0G used
```

## View the reason for down interfaces

When an interface goes down, it can take much time to find the reason using the logs and debug commands. Also, when a brief interface flap occurs, not all relevant information about what caused the flap is available later. To view the events that cause a physical or port-channel interface to go down, use the `show interface` and `show PortChannel` commands.

An interface flap can affect switch operation when performing time-consuming tasks with many database operations, such as forwarding database flush, MAC learning, Route delete, and Route add. To quickly troubleshoot an interface-down issue, refer to the reason in `show` command output.

### Use `show` commands to view the interface-down reason

Refer to the Reason column in `show interface status` output to see the high-level reason why a physical or port-channel interface is down.

- For physical interfaces, one of the following high-level reasons is displayed: `admin-down`, `err-disabled`, or `phy-link-down`. The reason `oper-up` indicates that the interface is up
- For port-channel interfaces, one of the following high-level reasons is displayed: `admin-down`, `min-links`, `err-disabled`, `all-links-down`, or `lacp-fail`. The reason `oper-up` indicates that the port-channel interface is up

```
sonic# show interface status

Name Description Oper Reason AutoNeg Speed MTU Alternative Name

Eth1/1 - down admin-down off 100000 9100 Ethernet0
Eth1/2/1 - down err-disabled off 10000 9100 Ethernet4
Eth1/2/2 - down phy-link-down off 10000 9100 Ethernet5
Eth1/2/3 - up oper-up off 10000 9100 Ethernet6
PortChannel1 - down admin-down - 20000 9100 -
PortChannel3 - down min-links - 30000 9100 -
PortChannel5 - down err-disabled - 30000 9100 -
PortChannel7 - down all-links-down - 40000 9100 -
PortChannel8 - down lacp-fail - 40000 9100 -
PortChannel9 - up oper-up - 10000 9100 -
```

To view the related event and timestamp when physical interfaces with a specific reason go down, use the `show interface status reason` command; for example:

```
sonic# show interface status err-disabled

Name Event Timestamp

Eth1/2/1 stp-status-up 2021-04-16 10:23:29
...
```

To view the sequence of events that led to a physical interface going down, use the `show interface Ethslot/port [.subport]` command; for example:

```
show interface Eth 1/48
Eth1/48 is up, line protocol is down, reason phy-link-down
Hardware is Eth, address is 18:5a:58:4f:ba:21
Mode of IPV4 address assignment: not-set
```

```

Mode of IPV6 address assignment: not-set
Interface IPv6 oper status: Disabled
IP MTU 9100 bytes
LineSpeed 1GB, Auto-negotiation off
FEC: DISABLED
Events:
 initialized at 2022-03-23T16:14:32.668561Z
 admin-up at 2022-03-23T16:14:33.88121Z
 xcvr-status-down at 2022-03-23T16:15:36.046996Z
 transceiver-incompatible at 2022-03-23T16:15:36.049866Z
 xcvr-status-up at 2022-03-23T16:15:36.052822Z
 port-enabled at 2022-03-23T16:15:36.055144Z
Last clearing of "show interface" counters: never
10 seconds input rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
10 seconds output rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
Input statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize
 0 Packets (128 to 255 Octects)
Output statistics:
 0 packets, 0 octets
 0 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded, 0 Oversize

```

Some of the events that may appear in the show command output are:

|                      |                          |                  |
|----------------------|--------------------------|------------------|
| admin-down           | speed-failed             | stp-status-down  |
| remote-fault         | if-type-failed           | stp-status-up    |
| local-fault          | media-type-failed        | xcvr-status-down |
| link-training-failed | incompatible-transceiver | xcvr-status-up   |
| pcs-errors           | transceiver-not-present  | udld-status-down |
| preempasis-failed    | port-breakout-in-progres | udld-status-up   |
| fec-failed           | port-breakout-completed  | phy-link-up      |

To view the sequence of events that led to a port-channel interface going down, use the `show interface PortChannel portchannel-number` command; for example:

```

sonic# show interface PortChannel 1
PortChannell is up, line protocol is up, reason oper-up, mode LACP
Hardware is PortChannel, address is 90:3c:b3:c5:bd:95
Minimum number of links to bring PortChannel up is 1
Mode of IPV4 address assignment: not-set
Mode of IPV6 address assignment: not-set
Fallback: Disabled
Graceful shutdown: Disabled
MTU 9100
LineSpeed 20.0GB
Events:
 all-links-down at 2021-10-21.02:53:15.567432
 lacp-fail at 2021-10-21.02:53:15.614807
 portchannel-up at 2021-10-21.02:53:19.077914
LACP mode ACTIVE interval SLOW priority 65535 address 90:3c:b3:c5:bd:95
Members in this channel: Ethernet0(Selected)
LACP Actor port 1 address 90:3c:b3:c5:bd:95 key 1
LACP Partner port 1 address 90:3c:b3:c5:95:bd key 1
Members in this channel: Ethernet1(Selected)
LACP Actor port 2 address 90:3c:b3:c5:bd:95 key 1
LACP Partner port 2 address 90:3c:b3:c5:95:bd key 1
Last clearing of "show interface" counters: never
10 seconds input rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
10 seconds output rate 0 packets/sec, 0 bits/sec, 0 Bytes/sec
Input statistics:
 30 packets, 5594 octets
 16 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded
Output statistics:
 42 packets, 6563 octets
 28 Multicasts, 0 Broadcasts, 0 Unicasts
 0 error, 0 discarded

```

## Use REST API to view the interface-down reason: Physical interfaces

You can also use a REST API Get request to retrieve physical interface status and view interface-down reasons. The returned reason can be OPER\_UP, PHY\_LINK\_DOWN, ERR\_DISABLED, or ADMIN\_DOWN. For example:

- GET /restconf/data/openconfig-interfaces:interfaces/interface={name}/openconfig-if-ethernet:etherent/state/openconfiginterfaces-ext:status/down-reason

Sample response:

```
{
 "openconfig-interfaces-ext:down-reason": "OPER_UP"
}
```

- GET /restconf/data/openconfig-interfaces:interfaces/interface={name}/openconfig-if-ethernet:etherent/openconfiginterfaces-ext:reason-events

Sample OPER\_UP response:

```
{
 "openconfig-interfaces-ext:reason-events": {
 "reason-event": [
 {
 "reason-event": {
 "reason": "OPER_UP",
 "event": "phy-link-up",
 "timestamp": "2021-06-06 09:29:55.639018"
 }
 }
]
 }
}
```

Sample PHY\_LINK\_DOWN response:

```
{
 "openconfig-interfaces-ext:reason-events": {
 "reason-event": [
 {
 "reason-event": {
 "reason": "PHY_LINK_DOWN",
 "event": "remote-fault",
 "timestamp": "2021-06-07 04:17:13.456626"
 }
 }
]
 }
}
```

## Use REST API to view the interface-down reason: Port-channel interfaces

Use a REST API Get request to retrieve port-channel interface status and view interface-down reasons. The returned reason can be OPER\_UP, ALL\_LINKS\_DOWN, ERR\_DISABLED, LACP\_FAIL, MIN\_LINKS, or ADMIN\_DOWN. For example:

- GET /restconf/data/openconfig-interfaces:interfaces/interface={name}/openconfig-if-aggregate:aggregation/state/openconfig-interfaces-ext:reason

Sample response:

```
{
 "openconfig-interfaces-ext:down-reason": "OPER_UP"
}
```

- GET /restconf/data/openconfig-interfaces:interfaces/interface={name}/openconfig-if-aggregate:aggregation/openconfiginterfaces-ext:reason-events

Sample ALL\_LINKS\_DOWN response:

```
{
 "openconfig-interfaces-ext:reason-events": {
 "reason-event": [
 {
 "state": {
 "event": "All-links-down",
 "reason": "ALL_LINKS_DOWN",
 "timestamp": "2021-08-30 05:53:31.432785"
 }
 }
]
 }
}
```

Sample ERR\_DISABLED response:

```
{
 "openconfig-interfaces-ext:reason-events": {
 "reason-event": [
 {
 "state": {
 "event": "delay_restore_status_down",
 "reason": "ERR_DISABLED",
 "timestamp": "2021-06-09 04:47:33.410626"
 }
 }
]
 }
}
```

## System reboot reason

If the switch is restarted for any reason, the system captures the event. You can view the reason for the last system restart using a `show` command.

The mechanism to capture the reboot reason is enabled on the system by default. You cannot disable it.

### View the reason for the last system reboot

```
sonic# show reboot-cause
```

#### Examples

System was restarted due to power loss:

```
sonic# show reboot-cause
Power Loss (Power on reset)
```

System was restarted due to the hardware watchdog:

```
sonic# show reboot-cause
Hardware Watchdog Reset
```

System was restarted after entering the `reboot` command:

```
sonic# show reboot-cause
User issued 'reboot' command [User: admin, Time: Wed 23 Mar 2022 04:01:15 PM UTC]
```

System was restarted for an unknown reason:

```
sonic# show reboot-cause
Unknown
```

# Unreliable Loss of Signal

For installed 10G and 40G SR/LR transceivers, improper signal tuning can result in a link-down or link-up state with many RX frame check sequence (FCS) errors because the electrical signal is not properly recovered. Enterprise SONiC provides unreliable Loss of Signal (LOS) support to avoid mistuned signal issues with installed transceivers. Automatic unreliable LOS detection is enabled by default.

**(i) NOTE:** Unreliable LOS is supported only on the S5200-ON series and Z9264F-ON switches.

**(i) NOTE:** When unreliable LOS is enabled, link tuning problems are prevented in most cases. Proper operation is not guaranteed.

**(i) NOTE:** Auto unreliable LOS enables unreliable LOS only on these 40G transceivers: FINISAR CORP FTL410QE2C, QSFP+ 40GBASE-SR4; FS QSFP-BD-40G, QSFP+ 40GBASE-SR-BiDi; FS QSFP-SR4-40G, and QSFP+ 40GBASE-SR4.

## Configure unreliable LOS

```
sonic(config)# interface Eth1/2
sonic(conf-if-Eth1/2)# unreliable-los {auto | on | off}
```

- auto — Automatically enables or disables unreliable LOS based on the transceiver that is detected on a port (default).
- on — Enables LOS detection.
- off — Disables LOS detection.

**(i) NOTE:** The no unreliable-los command does not remove the configuration from the interface. To reset unreliable LOS operation to the default auto mode, enter the `unreliable-los auto` command.

**(i) NOTE:** Dell Technologies recommends that you enable unreliable LOS on transceivers that do not have CDR support for 40G and 10G optics and active optical cables (AOCs), and 4x10G and 8x10G AOC breakout cables. To check if an installed optic or AOC has CDR support, enter the `show interfaces transceiver eeprom Ethernetport-number` command from the Linux shell; for example, for a 40G optic with CDR:

```
admin@sonic:~$ show interfaces transceiver eeprom Ethernet32
Ethernet32: SFP EEPROM detected
 Connector: LC
 Encoding: 64B66B
 Extended Identifier: Power Class 4(3.5W max), CDR present in Rx Tx
 Extended RateSelect Compliance: QSFP+ Rate Select Version 1
 Identifier: QSFP+
 Length OM3 (2m): 0.0
 Nominal Bit Rate(100Mbs): 206
 Specification compliance:
 10/40G Ethernet Compliance Code: N/A
 Gigabit Ethernet Compliant codes: 1000BASE-SX
 Vendor Date Code(YYYY-MM-DD Lot): 2020-07-30
 Vendor Name: DELL EMC
 Vendor OUI: 00-17-6A
 Vendor PN: XW7J0
 Vendor Rev: A0
 Vendor SN: AM07US0UVK
```

## View per-port unreliable LOS operational status

```
sonic# show interface unreliable-los status [Ethslot/port[/breakout-port] | port-range]
```

```
sonic# show interface unreliable-los status Eth 1/42-1/44
```

| Interface | Type              | Oper | Admin | If-State   |
|-----------|-------------------|------|-------|------------|
| Eth1/42   | QSFP+ 40GBASE-SR4 | on   | auto  | oper-up    |
| Eth1/43   |                   | off  | none  | admin-down |
| Eth1/44   | QSFP+ 40GBASE-SR4 | off  | auto  | oper-up    |

# Transceiver and cable diagnostics

This section describes the overview of transceiver and cable diagnostics and provides commands to view the diagnostics information.

## Transceiver diagnostics using DOM

Transceiver digital optical monitoring (DOM) information is a technology which allows you to monitor important parameters of the transceiver module in real-time. You can use DOM to monitor the TX (transmit) and RX (receive) ports of the module, as well as input and output power, temperature, and voltage. According to these monitored parameters, network technicians are able to check and ensure that the module is functioning well.

### View DOM summary

| Interface | Type    | Vendor        | Temp (C) | Volt (V) | RxPwr (dBm) | TxPwr (dBm) |
|-----------|---------|---------------|----------|----------|-------------|-------------|
| Eth1/1    | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/2    | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/3    |         |               |          |          |             |             |
| Eth1/4    | QSFP-DD | DELL          | 15.70    | 3.23     | -3.16       | -4.78       |
| Eth1/5    |         |               |          |          |             |             |
| Eth1/6    |         |               |          |          |             |             |
| Eth1/7    |         |               |          |          |             |             |
| Eth1/8    | QSFP-DD | DELL          | 13.83    | 3.28     | -40.00      | -40.00      |
| Eth1/9    | QSFP-DD | DELL          | 22.05    | N/A      | N/A         | N/A         |
| Eth1/10   |         |               |          |          |             |             |
| Eth1/11   | QSFP-DD | DELL          | 26.45    | N/A      | N/A         | N/A         |
| Eth1/12   | QSFP-DD | DELL          | 26.34    | N/A      | N/A         | N/A         |
| Eth1/13   |         |               |          |          |             |             |
| Eth1/14   |         |               |          |          |             |             |
| Eth1/15   | QSFP28  | FINISAR CORP  | 16.70    | 3.25     | -25.53      | -24.95      |
| Eth1/16/1 | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/16/2 | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/16/3 | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/16/4 | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/16/5 | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/16/6 | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/16/7 | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/16/8 | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/17   | QSFP-DD | DELL          | 29.05    | 3.23     | -40.00      | 1.76        |
| Eth1/18   | QSFP28  | DELL          | N/A      | N/A      | N/A         |             |
| Eth1/19   | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/20   | QSFP-DD | DELL          | 25.69    | N/A      | N/A         | N/A         |
| Eth1/21   | QSFP-DD | DELL          | 23.95    | N/A      | N/A         | N/A         |
| Eth1/22   | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/23   |         |               |          |          |             |             |
| Eth1/24   |         |               |          |          |             |             |
| Eth1/25   | QSFP-DD | DELL          | 29.01    | 3.23     | -40.00      | -40.00      |
| Eth1/26   | QSFP28  | DELL          | N/A      | N/A      | N/A         |             |
| Eth1/27   | QSFP-DD |               | 0.00     | 0.00     | -inf        |             |
| Eth1/28   | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/29   | QSFP-DD | DELL          | N/A      | N/A      | N/A         | N/A         |
| Eth1/30   | QSFP28  | DELL          | N/A      | N/A      | N/A         |             |
| Eth1/31   | QSFP-DD | DELL          | 23.45    | 3.22     | 2.89        | 2.30        |
| Eth1/32   |         |               |          |          |             |             |
| Eth1/33   | SFP     | FINISAR CORP. | 12.86    | 3.24     | -6.40       | -5.42       |

**(i) NOTE:** To view the show output in one display without having to page through screen displays, enter `show interface transceiver dom | no-more`.

| Interface | Type | Vendor | Temp (C) | Volt (V) | RxPwr (dBm) | TxPwr (dBm) |
|-----------|------|--------|----------|----------|-------------|-------------|
| Eth 1/5   |      |        |          |          |             |             |

```

Eth1/5 SFP FINISAR CORP. 28.4766 3.3018 -30.9691 -2.1488

```

```
sonic# show interface transceiver dom summary Eth 1/5-1/12
```

| Interface | Type | Vendor        | Temp (C) | Volt (V) | RxPwr (dBm) | TxPwr (dBm) |
|-----------|------|---------------|----------|----------|-------------|-------------|
| Eth1/5    | SFP  | FINISAR CORP. | 28.4766  | 3.3018   | -30.9691    | -2.1488     |
| Eth1/6    | SFP  | FINISAR CORP. | 30.0156  | 3.2948   | -33.9794    | -1.9199     |
| Eth1/7    | SFP  | JDSU          | 27.0625  | 3.3053   | -29.5861    | -4.9026     |
| Eth1/8    | SFP  | FINISAR CORP. | 0.0000   | 0.0000   | -inf        | -inf        |
| Eth1/9    | SFP  | FINISAR CORP. | 0.0000   | 0.0000   | -inf        | -inf        |
| Eth1/10   | SFP  | FINISAR CORP. | 30.9141  | 3.3026   | -36.9897    | -2.0149     |
| Eth1/11   | SFP  | FINISAR CORP. | 30.6602  | 3.2969   | -inf        | -1.9091     |
| Eth1/12   | SFP  | FINISAR CORP. | 28.9844  | 3.3443   | -33.9794    | -2.1375     |

To view the DOM and threshold information:

```
sonic# show interface transceiver Eth 1/10 dom
```

```

Eth1/10

```

```
Identifier: SFP
Vendor Name: DELL
Vendor Part: 3P3PG
ModuleMonitorValues:
 Temperature: 24.87 C
 Vcc: 3.30 Volts
ChannelMonitorValues:
 Rx1Power: -1.70 dBm
 Tx1Bias: 6.76 mA
 Tx1Power: -1.61 dBm
ChannelThresholdValues:
 RxPowerHighAlarm : 3.40 dBm
 RxPowerHighWarning: 2.40 dBm
 RxPowerLowAlarm : -10.50 dBm
 RxPowerLowWarning : -9.50 dBm
 TxBiasHighAlarm : 12.00 mA
 TxBiasHighWarning : 10.00 mA
 TxBiasLowAlarm : 0.00 mA
 TxBiasLowWarning : 0.00 mA
 TxPowerHighAlarm : 2.90 dBm
 TxPowerHighWarning: 2.40 dBm
 TxPowerLowAlarm : -8.60 dBm
 TxPowerLowWarning : -8.10 dBm
ModuleThresholdValues:
 TempHighAlarm : 80.00 C
 TempHighWarning: 75.00 C
 TempLowAlarm : -10.00 C
 TempLowWarning : -5.00 C
 VccHighAlarm : 3.60 Volts
 VccHighWarning : 3.50 Volts
 VccLowAlarm : 3.00 Volts
 VccLowWarning : 3.10 Volts
```

To view the DOM and threshold information without attached transceiver:

```
sonic# show interface transceiver Eth 1/5 dom
```

```

Eth 1/5

```

```
None detected
```

To view the DOM and threshold information of a QSFP-DD transceiver without DOM support:

```
sonic# show interface transceiver dom Eth 1/51/1
```

```

Eth1/51/1

```

```
Identifier: QSFP-DD
Vendor Name: DELL
Vendor Part: 229KM
DOM is not supported
```

To view the DOM and threshold information on all ports:

```
sonic# show interface transceiver dom
```

```

Eth 1/1

```

```
None detected
```

```

Eth 1/2

```

```
None detected
```

```

Eth 1/3

```

```
Identifier: QSFP-DD
Vendor Name: FIT HON TENG
Vendor Part: CU4EP54-01000-EF
DOM is not supported
```

```
..... omitted
```

```

Eth 1/31

```

```
Identifier: QSFP-DD
Vendor Name: AVAGO
Vendor Part: AFCT-93DRPHZ-AZ2
ChannelMonitorValues:
 Rx1Power: -33.9794 dBm
 Rx2Power: -23.2790 dBm
 Rx3Power: -32.2185 dBm
 Rx4Power: -22.8400 dBm
 Rx5Power: -inf dBm
 Rx6Power: -inf dBm
 Rx7Power: -inf dBm
 Rx8Power: -inf dBm
 Tx1Bias: 34.9920 mA
 Tx2Bias: 37.4960 mA
 Tx3Bias: 54.9520 mA
 Tx4Bias: 37.4960 mA
 Tx5Bias: 0.0000 mA
 Tx6Bias: 0.0000 mA
 Tx7Bias: 0.0000 mA
 Tx8Bias: 0.0000 mA
 Tx1Power: 3.0425 dBm
 Tx2Power: 3.1135 dBm
 Tx3Power: 2.7960 dBm
 Tx4Power: 3.0880 dBm
 Tx5Power: -inf dBm
 Tx6Power: -inf dBm
 Tx7Power: -inf dBm
 Tx8Power: -inf dBm
ChannelThresholdValues:
 RxPowerHighAlarm : 7.5000 dBm
 RxPowerHighWarning: 4.4999 dBm
 RxPowerLowAlarm : -10.4001 dBm
 RxPowerLowWarning : -6.4016 dBm
 TxBiasHighAlarm : 75.0000 mA
 TxBiasHighWarning : 70.0000 mA
 TxBiasLowAlarm : 10.0000 mA
 TxBiasLowWarning : 15.0000 mA
 TxPowerHighAlarm : 6.9999 dBm
 TxPowerHighWarning: 3.9999 dBm
 TxPowerLowAlarm : -6.4016 dBm
 TxPowerLowWarning : -2.4003 dBm
```

```

ModuleMonitorValues:
 Temperature: 30.7773 C
 Vcc: 3.3027 Volts
ModuleThresholdValues:
 TempHighAlarm : 75.0000 C
 TempHighWarning: 70.0000 C
 TempLowAlarm : -5.0000 C
 TempLowWarning : 0.0000 C
 VccHighAlarm : 3.6300 Volts
 VccHighWarning : 3.4650 Volts
 VccLowAlarm : 2.9700 Volts
 VccLowWarning : 3.1350 Volts

..... omitted

Displaying the DOM and Threshold information of Eth 1/21-1/31

sonic# show interface transceiver dom Eth 1/21-1/31

Eth 1/21

Identifier: QSFP-DD
Vendor Name: FIT HON TENG
Vendor Part: CU4EP54-01000-EF
DOM is not supported

Eth 1/22

None detected

Eth 1/23

None detected

..... omitted

Eth 1/31

Identifier: QSFP-DD
Vendor Name: AVAGO
Vendor Part: AFCT-93DRPHZ-AZ2
ChannelMonitorValues:
 Rx1Power: -33.9794 dBm
 Rx2Power: -23.2790 dBm
 Rx3Power: -32.2185 dBm
 Rx4Power: -22.8400 dBm
 Rx5Power: -inf dBm
 Rx6Power: -inf dBm
 Rx7Power: -inf dBm
 Rx8Power: -inf dBm
 Tx1Bias: 34.9920 mA
 Tx2Bias: 37.4960 mA
 Tx3Bias: 54.9520 mA
 Tx4Bias: 37.4960 mA
 Tx5Bias: 0.0000 mA
 Tx6Bias: 0.0000 mA
 Tx7Bias: 0.0000 mA
 Tx8Bias: 0.0000 mA
 Tx1Power: 3.0425 dBm
 Tx2Power: 3.1135 dBm
 Tx3Power: 2.7960 dBm
 Tx4Power: 3.0880 dBm
 Tx5Power: -inf dBm
 Tx6Power: -inf dBm
 Tx7Power: -inf dBm
 Tx8Power: -inf dBm
ChannelThresholdValues:
 RxPowerHighAlarm : 7.5000 dBm
 RxPowerHighWarning: 4.4999 dBm
 RxPowerLowAlarm : -10.4001 dBm

```

```

RxPowerLowWarning : -6.4016 dBm
TxBiasHighAlarm : 75.0000 mA
TxBiasHighWarning : 70.0000 mA
TxBiasLowAlarm : 10.0000 mA
TxBiasLowWarning : 15.0000 mA
TxPowerHighAlarm : 6.9999 dBm
TxPowerHighWarning: 3.9999 dBm
TxPowerLowAlarm : -6.4016 dBm
TxPowerLowWarning : -2.4003 dBm
ModuleMonitorValues:
 Temperature: 30.7773 C
 Vcc: 3.3027 Volts
ModuleThresholdValues:
 TempHighAlarm : 75.0000 C
 TempHighWarning: 70.0000 C
 TempLowAlarm : -5.0000 C
 TempLowWarning : 0.0000 C
 VccHighAlarm : 3.6300 Volts
 VccHighWarning : 3.4650 Volts
 VccLowAlarm : 2.9700 Volts
 VccLowWarning : 3.1350 Volts
..... omitted

```

## Cable diagnostics using TDR

A time-domain reflectometer (TDR) is an electronic instrument used to determine the characteristics of electrical lines by observing reflected waveforms.

Based on the platform hardware, the supported port types are:

- **Native RJ45:** A port with both MAC and PHY mounted on the box; the connections can easily established by cat5e and cat6 Ethernet cables.
- **1G Copper SFP(1000BASE-T):** A PHY-less switch design; the front panel ports are cages that allow external PHYs on SFP modules to get connected and communicated with the MAC on the switch. The conducted cable-diagnostics test leverage the TDR hardware engine on the copper SFP for this operation.

### Enable cable diagnostics on a port

```
test cable-diagnostics [Eth slot/port]
```

```

sonic# test cable-diagnostics
!!WARNING!! This operation may cause disruption of traffic, continue? [y/N]:y
sonic# test cable-diagnostics Eth 1/1
!!WARNING!! This operation may cause disruption of traffic, continue? [y/N]:y
sonic# test cable-diagnostics Eth 1/1-1/9
!!WARNING!! This operation may cause disruption of traffic, continue? [y/N]:y

```

### View cable diagnostics

```
show cable-diagnostics report [Eth slot/port]
```

```
sonic# show cable-diagnostics report
```

| Interface | Type | Length | Result        | Status    | Timestamp            |
|-----------|------|--------|---------------|-----------|----------------------|
| Eth1/1    | XCVR |        | Not Supported | COMPLETED | 29-Apr-2021 08:11:39 |
| Eth1/2    | XCVR |        | Not Supported | COMPLETED | 29-Apr-2021 08:11:39 |
| Eth1/3    | XCVR |        | Not Supported | COMPLETED | 29-Apr-2021 08:11:40 |
| Eth1/4    | XCVR |        | Not Supported | COMPLETED | 29-Apr-2021 08:11:40 |
| Eth1/5    | TDR  |        | OPEN          | COMPLETED | 29-Apr-2021 08:11:40 |
| Eth1/6    | TDR  | < 50m  | OK            | COMPLETED | 29-Apr-2021 08:11:41 |
| Eth1/7    | TDR  | < 50m  | OK            | COMPLETED | 29-Apr-2021 08:11:41 |
| Eth1/8    | XCVR |        | Not Supported | COMPLETED | 29-Apr-2021 08:11:41 |

```
sonic# show cable-diagnostics report Eth 1/4
```

| Interface | Type | Length | Result | Status    | Timestamp            |
|-----------|------|--------|--------|-----------|----------------------|
| Eth1/4    | TDR  | < 50m  | OK     | COMPLETED | 29-Apr-2021 08:11:41 |

```
sonic# show cable-diagnostics report Eth 1/3-1/8
```

| Interface | Type | Length | Result        | Status    | Timestamp            |
|-----------|------|--------|---------------|-----------|----------------------|
| Eth1/3    | XCVR |        | Not Supported | COMPLETED | 29-Apr-2021 08:11:40 |
| Eth1/4    | XCVR |        | Not Supported | COMPLETED | 29-Apr-2021 08:11:40 |
| Eth1/5    | TDR  |        | OPEN          | COMPLETED | 29-Apr-2021 08:11:40 |
| Eth1/6    | TDR  | < 50m  | OK            | COMPLETED | 29-Apr-2021 08:11:41 |
| Eth1/7    | TDR  | < 50m  | OK            | COMPLETED | 29-Apr-2021 08:11:41 |
| Eth1/8    | XCVR |        | Not Supported | COMPLETED | 29-Apr-2021 08:11:41 |

### View the cable-length

```
show cable-diagnostics cable-length [Eth slot/port]
```

```
sonic# show cable-diagnostics cable-length
```

| Interface | Type | Length |
|-----------|------|--------|
| Eth1/1    | TDR  | < 50m  |
| Eth1/2    | TDR  | < 50m  |
| Eth1/3    | TDR  |        |
| Eth1/4    | XCVR |        |

```
sonic# show cable-diagnostics cable-length Eth 1/2
```

| Interface | Type | Length |
|-----------|------|--------|
| Eth1/2    | TDR  | < 50m  |

```
sonic# show cable-diagnostics cable-length Eth 1/1-1/8
```

| Interface | Type | Length |
|-----------|------|--------|
| Eth1/1    | TDR  | < 50m  |
| Eth1/2    | TDR  | < 50m  |
| Eth1/3    | TDR  | < 50m  |
| Eth1/4    | TDR  | < 50m  |
| Eth1/5    | XCVR |        |
| Eth1/6    | XCVR |        |
| Eth1/7    | XCVR |        |
| Eth1/8    | XCVR |        |

# Advanced troubleshooting with Telemetry and Monitoring

Use Enterprise SONiC Telemetry and Monitoring (TAM) features to simplify the troubleshooting and monitoring of networking issues. TAM features take advantage of a switch's silicon capabilities. Some TAM use cases are:

- Ensure SLA conformance requiring end-to-end path latency for targeted flows.
- Identify switches and flows with traffic congestion.
- Identify patterns of network bandwidth usage.
- Analyze packet paths and path changes.
- Identify switches and flows with packet drops.
- Identify packet-drop patterns.

TAM features include:

- [Inband flow analytics](#) for flow monitoring
- [Packet drop monitoring](#)
- [Tailstamping](#)

To display the status of TAM features, use the `show tam features` command:

```
sonic# show tam features
Name Status
----- -----
ifa Active
drop-monitor Active
```

**i** **NOTE:** TAM features have different requirements for sampler rate and collector configuration.

**Table 40. TAM feature configuration — Sampler rate and collector**

| TAM feature                         | Sampler configuration required? | Collector configuration required? |
|-------------------------------------|---------------------------------|-----------------------------------|
| Inband flow analytics: Ingress node | Yes                             | No                                |
| Inband flow analytics: Transit node | No                              | No                                |
| Inband flow analytics: Egress node  | No                              | Yes                               |
| Packet drop monitoring              | Yes                             | Yes                               |
| Tailstamping                        | No                              | No                                |

## Topics:

- [Inband flow analytics](#)
- [Packet drop monitoring](#)
- [Tailstamping](#)

## Inband flow analytics

**i** **NOTE:** The Inband Flow Analyzer (IFA) is available only in the Cloud Premium and Enterprise Premium bundles. IFA is not available in the Cloud Standard, Enterprise Standard, and Edge Standard bundles.

The Inband Flow Analyzer records flow-specific information from switches across the network for specified flows. For detailed information about IFA operation, see the [Inband Flow Analyzer IETF draft](#).

The IFA protocol defines a header that marks the flow and directs the collection of analyzed metadata for marked packets per-hop across the network. IFA performs inband flow analysis, and possible actions on the flow data. After you configure a flow for IFA analysis, an ingress node makes a copy of the flow or samples the live traffic flow, or tags a live traffic flow for analysis and data collection. IFA copies a flow by packet sampling or by cloning the flow. The new, copied packets are representative packets of the original flow and possess the exact same characteristics as the original flow. As a result, IFA packets traverse the same path in the network and the same queues in the networking devices as the original packets.

The intermediate nodes keep appending their own metadata to the IFA-tagged packets before forwarding them. An egress node terminates the IFA flow by summarizing and extracting the metadata of the entire path and sending it to a configured collector. A network monitoring application on a Collector can analyze the IFA telemetry information and provide full network visibility, including metrics such as latency, packet loss, and the complete network path that a packet travels.

Using the IFA feature, you can set up monitoring sessions for specified flows. For a monitored flow, a switch is marked as an ingress node, a transit node, or a terminating node. A collector that receives the extracted metadata is identified by its IP address and transport parameters. Configure one or more collectors on terminating nodes.

## IFA configuration

To use IFA for telemetry and monitoring (TAM), you must enable IFA on the ingress, transit, and egress nodes in an IFA monitoring session. A switch's role is per-flow. The same switch can operate as the ingress device in one flow, and the transit device in another flow.

1. Configure the ingress node.

- a. Enter Telemetry and Monitoring configuration mode, and configure a 32-bit TAM switch identifier (1 to 4294967295; the default is the last 16-bits from the switch MAC address). The TAM ID uniquely identifies a switch for data flow monitoring. To delete a TAM ID, use the `no switch-id` command.

```
sonic(config)# tam
sonic(conf-tam)# switch-id id
```

**i | NOTE:** If any TAM features are active, any changes to the switch-wide global attributes do not take effect immediately.

- b. Configure a 32-bit Enterprise identifier that is used in telemetry reports (1 to 4294967295; the default is the last 16-bits from the switch MAC address). To delete an Enterprise ID, use the `no enterprise-id` command.

```
sonic(conf-tam)# enterprise-id id
```

**i | NOTE:** If any TAM features are active, any changes to the switch-wide global attributes do not take effect immediately.

- c. Create a sampler session (up to 63 characters) and configure the sampling rate — the number of packets in a flow out of which one packet is selected (1 to 4294967295).

```
sonic(conf-tam)# sampler name rate sampler_rate
```

- d. Configure a flow group with match criteria. The valid protocol values are UDP and TDP.

```
sonic(conf-tam)# flow-group name [src-ip source-ip-address/prefix-length] [dst-ip destination-ip-address/prefix-length] [src-14-port port-number] [dest-14-port port-number] [priority priority-value] [protocol protocol-value]
sonic(conf-tam)# exit
```

- e. Assign the flow group to an ingress Ethernet interface.

```
sonic(config)# interface Ethslot/port[/breakout-port]
sonic(config-if-Eth)# flow-group name
sonic(config-if-Eth)# exit
```

- f. Enable IFA on the ingress switch.

```
sonic(config)# tam
sonic(conf-tam)# ifa
sonic(config-tam-ifa)# enable
```

- g. Configure an IFA session for a specified flow group, including the sampler rate.

```
sonic(config-tam-ifa)# session session-name flowgroup flowgroup-name sampler sampler-name node-type ingress
```

```
sonic(config-tam-if) # exit
sonic(conf-tam) # exit
```

2. Configure each transit node.

- Enter Telemetry and Monitoring configuration mode, and configure a 32-bit TAM switch identifier (1 to 4294967295; the default is the last 16-bits from the switch MAC address).

```
sonic(config) # tam
sonic(conf-tam) # switch-id id
```

- Configure a 32-bit Enterprise identifier that is used in telemetry reports (1 to 4294967295; the default is the last 16-bits from the switch MAC address). To delete an Enterprise ID, use the no enterprise-id command.

```
sonic(conf-tam) # enterprise-id id
```

- Enable IFA on a transit switch.

```
sonic(conf-tam) # ifa
sonic(config-tam-if) # enable
sonic(config-tam-if) # exit
sonic(conf-tam) # exit
```

3. Configure the egress node with an attached collector.

- Enter Telemetry and Monitoring configuration mode, and configure the TAM switch identifier (1 to 4294967295; the default is the last 16-bits from the switch MAC address).

```
sonic(config) # tam
sonic(conf-tam) # switch-id id
```

- Configure a 32-bit Enterprise identifier that is used in telemetry reports (1 to 4294967295; the default is the last 16-bits from the switch MAC address). To delete an Enterprise ID, use the no enterprise-id command.

```
sonic(conf-tam) # enterprise-id id
```

- Configure the collector to which monitored TAM data is sent through the interface configured in either default VRF (without vrf option) or user-specific VRF (with vrf option). Reenter the command to configure additional collectors. The port number identifies the UDP/TCP port interface that receives TAM data. To delete a TAM collector, use the no collector-name command.

```
sonic(conf-tam) # collector name ip ip_address port {port-number [protocol protocol-type] [vrf vrf-name]}
```

- Specify the flow group with (optional) match criteria. The valid protocol values are UDP and TDP.

```
sonic(conf-tam) # flow-group name [src-ip source-ip-address] [dst-ip destination-ip-address] [src-l4-port port-number] [dest-l4-port port-number] [priority priority-value] [protocol protocol-value]
sonic(conf-tam) # exit
```

- Enable IFA on the egress switch.

```
sonic(config) # tam
sonic(conf-tam) # ifa
sonic(config-tam-if) # enable
```

- Configure the IFA session for a specified flow group on the egress node with the attached collector. Configure a collector by entering the collector name.

```
sonic(config-tam-if) # session session-name flowgroup flowgroup-name collector collector-name node-type egress
```

## View IFA configuration and status

```
sonic# show tam ifa
Status : Active
Version : 2.0
```

```
Switch ID : 2020
Enterprise ID : 674
```

```
sonic# show tam ifa sessions
Session : http_236 (Ingress)
Flow Group Name : tcp_port_236
Id : 4025
Priority : 100
SRC IP : 13.92.96.32
DST IP : 7.72.235.82
DST L4 Port : 236
Ingress Intf : Ethernet20
Collector : None
```

```
sonic# show tam samplers
Name Sample Rate
----- -----
s1 1
s34 1000
```

```
sonic# show tam flowgroups
Flow Group Name : f9
Id : 60
Priority : 100
SRC IP : 192.1.2.3/32
DST IP : 172.6.5.4/32
Packet Count : 5432

Flow Group Name : DEMO
Id : 1
Priority : 100
SRC IP : 1.1.1.1/32
DST IP : 4.4.4.4/32
Packet Count : 454
```

```
sonic# show tam collectors
Name IP Address Port Protocol
----- ----- -----
c2 2.2.2.2 7676 UDP
```

```
sonic# show tam switch
TAM Device information

Switch ID : 1234
Enterprise ID : 674
```

#### Example: Configure an IFA session

This example shows how to configure IFA to monitor all flows to a web server at 20.20.1.1. An IFA collector for analyzing the metadata is attached on a UDP port at 20.20.20.4:9090. The session monitors one packet out of every 1000 in a configured flow. Ingress flows are on Ethernet port 44 on switch1.

**i | NOTE:** This example does not cover the additional configuration required for the packet forwarding/routing.

```
; Configure ingress node
sonic(config)# tam
sonic(conf-tam)# switch-id 1234
sonic(conf-tam)# enterprise-id 4434
sonic(conf-tam)# sampler websamp rate 1000
sonic(conf-tam)# flow-group websrvflows dst-ip 20.20.1.1 dst-l4-port 80 protocol TCP
sonic(conf-tam)# exit
sonic(config)# interface Ethernet 44
sonic(config-if-Ethernet44)# flow-group websrvflows
sonic(config-if-Ethernet44)# exit
sonic(config)# tam
sonic(conf-tam)# ifa
sonic(config-tam-if-a)# enable
sonic(config-tam-if-a)# session webflowmonitor flowgroup websrvflows sample-rate websamp
```

```

node-type ingress

; Configure transit node
sonic(config)# tam
sonic(conf-tam)# switch-id 1235
sonic(conf-tam)# enterprise-id 4434
sonic(conf-tam)# ifa
sonic(config-tam-iffa)# enable

; Configure egress node
sonic(config)# tam
sonic(conf-tam)# switch-id 1236
sonic(conf-tam)# enterprise-id 4434
sonic(conf-tam)# collector ifacoll1 ip 20.20.20.4 port 9090 protocol UDP
sonic(conf-tam)# collector dm_col ip 192.168.9.2 port 2055 vrf VRF_Red
sonic(conf-tam)# flow-group websrvflows dst-ip 20.20.1.1 dst-l4-port 80 protocol TCP
sonic(conf-tam)# ifa
sonic(config-tam-iffa)# enable
sonic(config-tam-iffa)# session webflowmonitor flowgroup websrvflows collector ifacoll1
nodetype egress

```

## Packet drop monitoring

**(i) NOTE:** Packet drop monitoring is available only in the Cloud Premium and Enterprise Premium bundles. Packet drop monitoring is not available in the Cloud Standard, Enterprise Standard, and Edge Standard bundles.

To receive reports on the packet drops in a flow, you can configure packet-drop monitoring sessions for specific flows on a switch. Configure a collector on a destination port to which dropped-packet reports are sent.

Packet drops are detected and reported on a per-flow basis to an external collector. Dropped packets in a flow are sampled using the configured sampling rate.

Configure a flow group with match criteria for monitoring drops. Dropped-packet reports that are sent to the collector in protobuf format include these drop events:

- Drop start — Sent when packet drops are observed for the first time in a flow. The drop event, flow keys, first 128 bytes of the packet, and the last observed drop reasons are reported.
- Drop active — Sent whenever the drop reason in a flow changes. The drop event, flow keys, and the drop reasons are reported.
- Drop stop — Sent when no sampled dropped packets are observed for the aging interval. The last observed drop reasons, drop event, and flow keys are reported.

### Configure packet drop monitoring

1. Enter Telemetry and Monitoring configuration mode, and configure the TAM switch identifier (1 to 4294967295; the default is the last 16-bits from the switch MAC address). The TAM ID uniquely identifies a switch for data flow monitoring. To delete a TAM ID, use the `no switch-id` command.

```

sonic(config) # tam
sonic(conf-tam) # switch-id id

```

2. Create a sampler session (up to 63 characters) and configure the sampling rate — the number of packets in a flow out of which one packet is selected (1 to 4294967295).

```

sonic(conf-tam) # sampler name rate sampler_rate

```

3. Configure a flow group with match criteria. The valid protocol values are UDP and TDP.

```

sonic(conf-tam) # flow-group name [src-ip source-ip-address] [dst-ip destination-ip-address] [src-l4-port port-number] [dest-l4-port port-number] [priority priority-value] [protocol protocol-value]
sonic(conf-tam) # exit

```

4. Assign the flow group to an ingress Ethernet interface.

```

sonic(config) # interface Ethslot/port[/breakout-port]
sonic(config-if-Eth) # flow-group name
sonic(config-if-Eth) # exit

```

5. Configure the collector to which dropped-packet reports are sent through the interface configured in either default VRF (without `vrf` option) or user-specific VRF (with `vrf` option). Reenter the command to configure additional collectors. The port number identifies the UDP/TCP port interface that receives TAM data. To delete a TAM collector, use the `no collector-name` command.

```
sonic(conf-tam)# collector name ip ip_address port {port-number [protocol protocol-type] [vrf Vrf-name]}
sonic(conf-tam)# exit
```

6. Enable packet drop monitoring on the switch.

```
sonic(config)# tam
sonic(conf-tam)# drop-monitor
sonic(config-tam-dm)# enable
```

7. Configure the aging interval for how long packet-drop data is recorded for a specified flow in a session. To delete a configured aging time, enter the `no aging-interval` command.

```
sonic(config-tam-dm)# aging-interval seconds
```

8. Configure a packet-drop monitoring session for a specified flow group. Configure a sampler rate and a collector by entering the collector name.

```
sonic(config-tam-dm)# session session-name flowgroup flowgroup-name collector
collector-name sampler sampler-name node-type ingress
```

#### **View packet-drop monitoring configuration and status**

```
sonic# show tam drop-monitor
Status : Active
Switch ID : 2020
Aging Interval : 20
```

| Name | Flow Group | Collector | Sampler |
|------|------------|-----------|---------|
| ss1  | f1         | c1        | s1      |
| ss2  | DEMO       | c1        | s2      |
| ss91 | f9         | c2        | s1      |

```
sonic # show tam drop-monitor sessions ss1
Session : ss1
Flow Group Name : f1
Id : 4025
Priority : 100
SRC IP : 13.92.96.32
DST IP : 7.72.235.82
DST L4 Port : 236
Ingress Intf : Ethernet20
Collector : c1
Sampler : s1
Packet Count : 7656
```

```
sonic# show tam samplers
Name Sample Rate

s1 1
s34 1000
```

```
sonic# show tam flowgroups
Flow Group Name : f9
Id : 60
Priority : 100
SRC IP : 192.1.2.3/32
DST IP : 172.6.5.4/32
Packet Count : 5432
```

```

Flow Group Name : DEMO
Id : 1
Priority : 100
SRC IP : 1.1.1.1/32
DST IP : 4.4.4.4/32
Packet Count : 454

```

| Name | IP Address | Port | Protocol |
|------|------------|------|----------|
| c2   | 2.2.2.2    | 7676 | UDP      |

### Example: Configure a packet-drop monitoring session

This example shows how to monitor all packet drops on a switch. All flows are destined to a web server at 20.20.1.1. A collector for analyzing the metadata is installed on a UDP port at 20.20.20.4:9091. The aging time for recording packet drops in the specified flow in the session is 5 seconds. The session monitors one packet out of every 100 in ingress flows on Ethernet port 44 on switch1.

**(i) NOTE:** This example does not cover the additional configuration required for the packet forwarding/routing.

```

sonic(config)# tam
sonic(conf-tam)# switch-id 1234
sonic(conf-tam)# sampler websamp rate 100
sonic(conf-tam)# flow-group websrvflows dst-ip 20.20.1.1 dst-l4-port 80 protocol TCP
sonic(conf-tam)# exit
sonic(config)# interface Ethernet 44
sonic(config-if-Ethernet44)# flow-group websrvflows
sonic(config-if-Ethernet44)# exit
sonic(config)# tam
sonic(conf-tam)# collector ifacoll1 ip 20.20.20.4 port 9090 protocol UDP
sonic(conf-tam)# collector mod_col ip 192.168.9.2 port 2055 vrf VRF_Blue
sonic(conf-tam)# drop-monitor
sonic(config-tam-dm)# enable
sonic(config-tam-dm)# aging-interval 5
sonic(config-tam-dm)# session webflowmonitor flowgroup websrvflows sample-rate websamp
collector dmcoll

```

### Packet drop report

The packet-drop reports sent to a collector are coded in protobuf format; for example:

```

/*
 * Copyright (c) 2017 Broadcom. The term "Broadcom" refers
 * to Broadcom Limited and/or its subsidiaries.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 * http://www.apache.org/licenses/LICENSE-2.0
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
syntax = "proto2";

message Event {
 required uint64 timestamp = 1;

 oneof EventType {
 MirrorOnDrop mod_event = 2;
 FlowLearning flow_event = 3;
 RouteChangeDetection rcd_event = 4;
 NetworkLatencyMonitoring nlm_event = 5;
 }
}

message MirrorOnDrop {
 enum MOD_STATE {
 DROP_INVALID = 0;

```

```

 DROP_START = 1;
 DROP_ACTIVE = 2;
 DROP_STOP = 3;
 }

 optional MOD_STATE mod_state = 1;
}

message FlowLearning {
 enum FL_STATE {
 FL_INVALID = 0;
 FL_LEARN = 1;
 FL_AGING = 2;
 FL_EXPORT = 3;
 FL_TABLE_FULL = 4; /* Atomic event, no other information need to be send */
 }
}

optional FL_STATE fl_state = 1;

message RouteChangeDetection {
 optional uint32 dummy = 1;
}

message NetworkLatencyMonitoring {
 optional uint32 dummy = 1;
}

message Flow {
 enum DROP_REASON {
 DROP_INVALID = 0;
 L3_SOURCE_MISS = 1;
 L3_DEST_MISS = 2;
 MCAST_MISS = 3;
 IP_MCAST_MISS = 4;
 UNKNOWN_VLAN = 5;
 L3_HEADER_MISMATCH = 6;
 DOS_ATTACK = 7;
 MARTIAN_ADDR = 8;
 TUNNEL_ERROR = 9;
 PARITY_ERROR = 10;
 L3_MTU_FAIL = 11;
 HIGIG_HDR_ERROR = 12;
 MCAST_IDX_ERROR = 13;
 CLASS_BASED_MOVE = 14;
 L3_ADDR_BIND_FAIL = 15;
 MPLS_LABEL_MISS = 16;
 MPLS_INVALID_ACTION = 17;
 MPLS_INVALID_PAYLOAD = 18;
 TUNNEL_OBJECT_VALIDATION_FAIL = 19;
 MPLS_SEQUENCE_NUMBER = 20;
 L2_NON_UNICAST_MISS = 21;
 NIV_PRIO_DROP = 22;
 NIV_RPF_FAIL = 23;
 UNKNOWN_SUBTENDING_PORT = 24;
 TUNNEL_ADAPT_LOOKUP_MISS = 25;
 PACKET_FLOW_SELECT_MISS = 26;
 TUNNEL_DECAP_ECN_ERROR = 27;
 FAILOVER_DROP = 28;
 OTHER_LOOKUP_MISS = 29;
 INVALID_TPID = 30;
 TUNNEL_TTL_ERROR = 31;
 MPLS_ILLEGAL_RESERVED_LABEL = 32;
 L3_HEADER_ERROR = 33;
 L2_HEADER_ERROR = 34;
 TTL1 = 35;
 TTL = 36;
 FCOE_ZONE_CHECK_FAIL = 37;
 IPMC_INTERFACE_MISMATCH = 38;
 MPLS_TTL = 39;
 MPLS_UNKNOWN_ACH = 40;
 OAM_ERROR = 41;
 }
}

```

```

L2_GRE_SIP_MISS = 42;
L2_GRE_VPNID_MISS = 43;
BFD_ERROR = 44;
CONGESTION_CNM_PROXY_ERROR = 45;
VXLAN_SIP_MISS = 46;
VXLAN_VPNID_MISS = 47;
NIV_INTERFACE_MISS = 48;
NIV_TAG_INVALID = 49;
NIV_TAG_DROP = 50;
NIV_UNTAG_DROP = 51;
TRILL_INVALID = 52;
TRILL_MISS = 53;
TRILL_TTL = 55;
NAT_ERROR = 56;
TCP_UDP_NAT_MISS = 57;
ICMP_NAT_MISS = 58;
NAT_FRAGMENT = 59;
NAT_MISS = 60;
}

optional uint32 proto = 1;
optional uint32 sip = 2;
optional uint32 dip = 3;
optional uint32 l4_sport = 4;
optional uint32 l4_dport = 5;
optional uint32 vnid = 6;
optional uint32 inner_proto = 7;
optional uint32 inner_sip = 8;
optional uint32 inner_dip = 9;
optional uint32 inner_l4_sport = 10;
optional uint32 inner_l4_dport = 11;
optional bytes custom_key = 12;
optional uint32 group_id = 13; /* Packet type is derived directly from
group ID */
optional bytes packet = 14;
optional uint64 packet_count = 15;
optional uint64 byte_count = 16;
optional DROP_REASON drop_reason_1 = 17;
optional DROP_REASON drop_reason_2 = 18;
}

message EventPair {
 repeated Event event = 1;
 repeated Flow flow = 2;
}

message GenEvent {
 required string system_id = 1;
 optional uint32 component_id = 2;
 optional uint32 sub_component_id = 3;
 repeated EventPair eventpair = 4;
 optional string hostname = 5;
}

```

## Tailstamping

**i** **NOTE:** Tailstamping is available only in the Cloud Premium and Enterprise Premium bundles. Tailstamping is not available in the Cloud Standard, Enterprise Standard, and Edge Standard bundles.

The tailstamping feature attaches arrival and departure timestamps with a switch identifier to a frame at one or more switches along a flow path across the network. Enable tailstamping for a set of flows that match configured criteria.

Use a collector to gather and analyze timestamps. Use a network monitoring application to analyze the timestamps and measure the latency of a flow across the network, trace packets across a path, detect hotspot points, and validate packet arrival sequence.

### Configure tailstamping

- Enter Telemetry and Monitoring configuration mode, and configure the TAM switch identifier (1 to 4294967295; the default is last 16-bits from the switch MAC address). The TAM ID uniquely identifies a switch for data flow monitoring. To delete a TAM ID, use the no switch-id command.

```
sonic(config)# tam
sonic(conf-tam)# switch-id id
```

- Configure a flow group with match criteria. The valid protocol values are UDP and TDP.

```
sonic(conf-tam)# flow-group name [src-ip source-ip-address] [dst-ip destination-ip-
address] [src-l4-port port-number] [dest-l4-port port-number] [priority priority-
value] [protocol protocol-value]
sonic(conf-tam)# exit
```

- Enable tailstamping on the switch.

```
sonic(conf-tam)# tail-stamping
sonic(config-tam-ts)# enable
```

- Configure a tailstamping session for a specified flow group.

```
sonic(config-tam-ts)# session session-name flowgroup flowgroup-name
```

#### **View tailstamping configuration and status**

```
sonic# show tail-stamping

Status : Active
Switch ID : 2020

sonic # show tam tail-stamping sessions

Name Flow Group
----- -----
http_236 tcp_port_236
http_239 tcp_port_239
http_241 tcp_port_241
```

```
sonic # show tam tail-stamping sessions http_236

Session : http_236
Flow Group Name: tcp_port_236
Id : 4025
Priority : 100
SRC IP : 13.92.96.32
DST IP : 7.72.235.82
DST L4 Port : 236
Packet Count : 7656
```

#### **Example: Configure a tailstamping session**

This example shows how to gather flow metadata for network probe packets sent from 10.10.1.1:8080 to a TCP port at 20.4.5.2:7070.

**(i) NOTE:** This example does not cover the additional configuration required for the packet forwarding/routing.

```
sonic(config)# tam
sonic(conf-tam)# switch-id 1234
sonic(conf-tam)# flow-group probeflow src-ip 10.10.1.1 src-14-port 8080 dst-ip 20.4.5.2
dst-14-port 7070 protocol TCP
sonic(conf-tam)# tail-stamping
sonic(config-tam-ts)# enable
sonic(config-tam-ts)# session probemonitor flowgroup probeflow
```

## Support resources

The Dell Support site provides a range of documents and tools to assist you with effectively using Dell devices. Through the support site you can obtain technical information regarding Dell products, access software upgrades and patches, download available management software, and manage your open cases. The Dell support site provides integrated, secure access to these services.

To access the Dell Technologies Support site, go to [www.dell.com/support/](http://www.dell.com/support/). To display information in your language, scroll down to the bottom of the page and select your country or region from the drop-down menu.

- To obtain product-specific information, enter the 7-character service tag or 11-digit express service code of your switch and click **Submit**.
- To view the service tag or express service code, pull out the luggage tag on the chassis or enter the `show chassis` command from the CLI.
- To receive additional kinds of technical support, click **Contact Us**, then click **Technical Support**.

To access system documentation, see [www.dell.com/manuals/](http://www.dell.com/manuals/).

To search for drivers and downloads, see [www.dell.com/drivers/](http://www.dell.com/drivers/).

To participate in Dell community blogs and forums, see [www.dell.com/community](http://www.dell.com/community).

# MIB objects

Enterprise SONiC supports the following MIB objects:

## #BGP

```
bgpVersion 1.3.6.1.2.1.15.1
bgpLocalAs 1.3.6.1.2.1.15.2
bgpPeerTable 1.3.6.1.2.1.15.3
bgpIdentifier 1.3.6.1.2.1.15.4
bgp4PathAttrTable 1.3.6.1.2.1.15.6
```

## #BRIDGE-MIB

```
dot1dBaseBridgeAddress 1.3.6.1.2.1.17.1.1
dot1dBaseNumPorts 1.3.6.1.2.1.17.1.2
dot1dBaseType 1.3.6.1.2.1.17.1.3
dot1dBasePortTable 1.3.6.1.2.1.17.1.4
dot1dTpAgingTime 1.3.6.1.2.1.17.4.2
```

## #Dell-Vendor-MIB.mib

```
productIdentificationDisplayName 1.3.6.1.4.1.674.10895.3000.1.2.100.1
productIdentificationVendor 1.3.6.1.4.1.674.10895.3000.1.2.100.3
productIdentificationSerialNumber 1.3.6.1.4.1.674.10895.3000.1.2.100.8.1.2.1
productIdentificationServiceTag 1.3.6.1.4.1.674.10895.3000.1.2.100.8.1.4.1
```

## #DisManEventMIB

```
mteResourceSampleMinimum 1.3.6.1.2.1.88.1.1.1
mteResourceSampleInstanceMaximum 1.3.6.1.2.1.88.1.1.2
mteResourceSampleInstances 1.3.6.1.2.1.88.1.1.3
mteResourceSampleInstancesHigh 1.3.6.1.2.1.88.1.1.4
mteResourceSampleInstanceLacks 1.3.6.1.2.1.88.1.1.5
mteTriggerFailures 1.3.6.1.2.1.88.1.2.1
mteObjectsTable 1.3.6.1.2.1.88.1.3.1
mteEventTable 1.3.6.1.2.1.88.1.4.2
mteEventNotificationTable 1.3.6.1.2.1.88.1.4.3
```

## #ENTITY

```
entPhysicalTable 1.3.6.1.2.1.47.1.1.1
entPhySensorTable 1.3.6.1.2.1.99.1.1
entStateTable 1.3.6.1.2.1.131.1.1
```

## #HR-MIB

```
hrSystemUptime 1.3.6.1.2.1.25.1.1
hrSystemDate 1.3.6.1.2.1.25.1.2
hrSystemInitialLoadDevice 1.3.6.1.2.1.25.1.3
hrSystemInitialLoadParameters 1.3.6.1.2.1.25.1.4
hrSystemNumUsers 1.3.6.1.2.1.25.1.5
hrSystemProcesses 1.3.6.1.2.1.25.1.6
hrSystemMaxProcesses 1.3.6.1.2.1.25.1.7
hrMemorySize 1.3.6.1.2.1.25.2.2
hrStorageTable 1.3.6.1.2.1.25.2.3
hrDeviceTable 1.3.6.1.2.1.25.3.2
hrProcessorTable 1.3.6.1.2.1.25.3.3
hrFSTable 1.3.6.1.2.1.25.3.8
hrSWRunTable 1.3.6.1.2.1.25.4.2
hrSWRunPerfTable 1.3.6.1.2.1.25.5.1
hrSWInstalledTable 1.3.6.1.2.1.25.6.3
```

## #ICMP

```
sicmpInMsgs 1.3.6.1.2.1.5.1
icmpInErrors 1.3.6.1.2.1.5.2
icmpInDestUnreachs 1.3.6.1.2.1.5.3
icmpInTimeExclds 1.3.6.1.2.1.5.4
icmpInParmProbs 1.3.6.1.2.1.5.5
icmpInSrcQuenches 1.3.6.1.2.1.5.6
icmpInRedirects 1.3.6.1.2.1.5.7
icmpInEchos 1.3.6.1.2.1.5.8
icmpInEchoReps 1.3.6.1.2.1.5.9
icmpInTimestamps 1.3.6.1.2.1.5.10
icmpInTimestampReps 1.3.6.1.2.1.5.11
icmpInAddrMasks 1.3.6.1.2.1.5.12
icmpInAddrMaskReps 1.3.6.1.2.1.5.13
icmpOutMsgs 1.3.6.1.2.1.5.14
icmpOutErrors 1.3.6.1.2.1.5.15
icmpOutDestUnreachs 1.3.6.1.2.1.5.16
icmpOutTimeExclds 1.3.6.1.2.1.5.17
icmpOutParmProbs 1.3.6.1.2.1.5.18
icmpOutSrcQuenches 1.3.6.1.2.1.5.19
icmpOutRedirects 1.3.6.1.2.1.5.20
icmpOutEchos 1.3.6.1.2.1.5.21
icmpOutEchoReps 1.3.6.1.2.1.5.22
icmpOutTimestamps 1.3.6.1.2.1.5.23
icmpOutTimestampReps 1.3.6.1.2.1.5.24
icmpOutAddrMasks 1.3.6.1.2.1.5.25
icmpOutAddrMaskReps 1.3.6.1.2.1.5.26
icmpStatsTable 1.3.6.1.2.1.5.29
icmpMsgStatsTable 1.3.6.1.2.1.5.30
```

## #IF-MIB

```
ifNumber 1.3.6.1.2.1.2.1
ifTable 1.3.6.1.2.1.2.2
ifXTable 1.3.6.1.2.1.31.1.1
```

## #IP-MIB

```
ipForwarding 1.3.6.1.2.1.4.1
ipDefaultTTL 1.3.6.1.2.1.4.2
ipRouteNextHop 1.3.6.1.2.1.4.21.1.7
ipNetToMediaPhysAddress 1.3.6.1.2.1.4.22.1.2
ipForwardNumber 1.3.6.1.2.1.4.24
ipForwardTable 1.3.6.1.2.1.4.24.2
ipCidrRouteNumber 1.3.6.1.2.1.4.24.3
ipv6IpForwarding 1.3.6.1.2.1.4.25
ipv6IpDefaultHopLimit 1.3.6.1.2.1.4.26
ipSystemStatsTable 1.3.6.1.2.1.4.31.1
ipIfStatsTableLastChange 1.3.6.1.2.1.4.31.2
ipIfStatsTable 1.3.6.1.2.1.4.31.3
ipAddressPrefixTable 1.3.6.1.2.1.4.32
ipAddressSpinLock 1.3.6.1.2.1.4.33
ipAddressTable 1.3.6.1.2.1.4.34
ipNetToPhysicalTable 1.3.6.1.2.1.4.35
ipv6ScopeZoneIndexTable 1.3.6.1.2.1.4.36
ipDefaultRouterTable 1.3.6.1.2.1.4.37
```

## #LLDP-MIB

```
LLDPLocalSystemData1.0.8802.1.1.2.1.3
LLDPLocPortTable1.0.8802.1.1.2.1.3.7
LLDPLocManAddrTable1.0.8802.1.1.2.1.3.8
LLDPRemTable 1.0.8802.1.1.2.1.4.1
LLDPRemManAddrTable1.0.8802.1.1.2.1.4.2
```

## #net-snmp

```
memIndex .1.3.6.1.4.1.2021.4.1.0
memErrorName .1.3.6.1.4.1.2021.4.2.0
memTotalSwap .1.3.6.1.4.1.2021.4.3.0
```

```

memAvailSwap .1.3.6.1.4.1.2021.4.4.0
memTotalReal .1.3.6.1.4.1.2021.4.5.0
memAvailReal .1.3.6.1.4.1.2021.4.6.0
memTotalFree .1.3.6.1.4.1.2021.4.11.0
memMinimumSwap .1.3.6.1.4.1.2021.4.12.0
memShared .1.3.6.1.4.1.2021.4.13.0
memBuffer .1.3.6.1.4.1.2021.4.14.0
memCached .1.3.6.1.4.1.2021.4.15.0
memSwapError .1.3.6.1.4.1.2021.4.100.0
memSwapErrorMsg .1.3.6.1.4.1.2021.4.101.0
laTable 1.3.6.1.4.1.2021.10
systemStats 1.3.6.1.4.1.2021.11
diskIOTable 1.3.6.1.4.1.2021.13.15.1
lmTempSensorsTable 1.3.6.1.4.1.2021.13.16.2
lmFanSensorsTable 1.3.6.1.4.1.2021.13.16.3
nsModuleTable 1.3.6.1.4.1.8072.1.2.1.1.4
nsCacheDefaultTimeout 1.3.6.1.4.1.8072.1.5.1
nsCacheEnabled 1.3.6.1.4.1.8072.1.5.2
nsCacheTable 1.3.6.1.4.1.8072.1.5.3
nsDebugEnabled 1.3.6.1.4.1.8072.1.7.1.1
nsDebugOutputAll 1.3.6.1.4.1.8072.1.7.1.2
nsDebugDumpPdu 1.3.6.1.4.1.8072.1.7.1.3
nsLoggingTable 1.3.6.1.4.1.8072.1.7.2.1
nsVacmAccessTable 1.3.6.1.4.1.8072.1.9.1
cefFcFRUPowerOperStatus 1.3.6.1.4.1.9.9.117.1.1.2.1.2
csqIfQosGroupStatsValue 1.3.6.1.4.1.9.9.580.1.5.5.1.4
cpfcIfRequests 1.3.6.1.4.1.9.9.813.1.1.1.1
cpfcIfIndications 1.3.6.1.4.1.9.9.813.1.1.1.2
cpfcIfPriorityRequests 1.3.6.1.4.1.9.9.813.1.2.1.2
cpfcIfPriorityIndications 1.3.6.1.4.1.9.9.813.1.2.1.3

```

#### #notification log mib

```

nlmConfigGlobalEntryLimit 1.3.6.1.2.1.92.1.1.1
nlmConfigGlobalAgeOut 1.3.6.1.2.1.92.1.1.2
nlmStatsGlobalNotificationsLogged 1.3.6.1.2.1.92.1.2.1
nlmStatsGlobalNotificationsBumped 1.3.6.1.2.1.92.1.2.2

```

#### #OSPF

```

ospfRouterId 1.3.6.1.2.1.14.1.1.0
ospfAdminStat 1.3.6.1.2.1.14.1.2.0
ospfVersionNumber 1.3.6.1.2.1.14.1.3.0
ospfAreaBdrRtrStatus 1.3.6.1.2.1.14.1.4.0
ospfAsBdrRtrStatus 1.3.6.1.2.1.14.1.5.0
ospfExternLsaCount 1.3.6.1.2.1.14.1.6.0
ospfExternLsaCksumSum 1.3.6.1.2.1.14.1.7.0
ospfTOSSupport 1.3.6.1.2.1.14.1.8.0
ospfOriginateNewLsas 1.3.6.1.2.1.14.1.9.0
ospfRxNewLsas 1.3.6.1.2.1.14.1.10.0
ospfExtLsdbLimit 1.3.6.1.2.1.14.1.11.0
ospfMulticastExtensions 1.3.6.1.2.1.14.1.12.0
ospfExitOverflowInterval 1.3.6.1.2.1.14.1.13.0
ospfDemandExtensions 1.3.6.1.2.1.14.1.14.0
ospfAreaTable 1.3.6.1.2.1.14.2
ospfStubAreaTable 1.3.6.1.2.1.14.3
ospfLsdbTable 1.3.6.1.2.1.14.4
ospfAreaRangeTable 1.3.6.1.2.1.14.5
ospfIfTable 1.3.6.1.2.1.14.7
ospfIfMetricTable 1.3.6.1.2.1.14.8
ospfVirtIfTable 1.3.6.1.2.1.14.9
ospfNbrTable 1.3.6.1.2.1.14.10
ospfVirtNbrTable 1.3.6.1.2.1.14.11
ospfExtLsdbTable 1.3.6.1.2.1.14.12

```

#### #Q-BRIDGE-MIB

```

dot1qVlanVersionNumber 1.3.6.1.2.1.17.7.1.1.1
dot1qMaxVlanId 1.3.6.1.2.1.17.7.1.1.2
dot1qMaxSupportedVlans 1.3.6.1.2.1.17.7.1.1.3
dot1qNumVlans 1.3.6.1.2.1.17.7.1.1.4
dot1qFdbTable 1.3.6.1.2.1.17.7.1.2.1

```

```
dot1qTpFdbTable 1.3.6.1.2.1.17.7.1.2.2
dot1qVlanCurrentTable 1.3.6.1.2.1.17.7.1.4.2
dot1qVlanStaticTable 1.3.6.1.2.1.17.7.1.4.3
dot1qPvid 1.3.6.1.2.1.17.7.1.4.5.1.1
```

## #SNMP

```
snmpInPkts 1.3.6.1.2.1.11.1
snmpOutPkts 1.3.6.1.2.1.11.2
snmpInBadVersions 1.3.6.1.2.1.11.3
snmpInBadCommunityNames 1.3.6.1.2.1.11.4
snmpInBadCommunityUses 1.3.6.1.2.1.11.5
snmpInASNParseErrs 1.3.6.1.2.1.11.6
snmpInTooBigs 1.3.6.1.2.1.11.8
snmpInNoSuchNames 1.3.6.1.2.1.11.9
snmpInBadValues 1.3.6.1.2.1.11.10
snmpInReadOnlys 1.3.6.1.2.1.11.11
snmpInGenErrs 1.3.6.1.2.1.11.12
snmpInTotalReqVars 1.3.6.1.2.1.11.13
snmpInTotalSetVars 1.3.6.1.2.1.11.14
snmpInGetRequests 1.3.6.1.2.1.11.15
snmpInGetNexsts 1.3.6.1.2.1.11.16
snmpInSetRequests 1.3.6.1.2.1.11.17
snmpInGetResponses 1.3.6.1.2.1.11.18
snmpInTraps 1.3.6.1.2.1.11.19
snmpOutTooBigs 1.3.6.1.2.1.11.20
snmpOutNoSuchNames 1.3.6.1.2.1.11.21
snmpOutBadValues 1.3.6.1.2.1.11.22
snmpOutGenErrs 1.3.6.1.2.1.11.24
snmpOutGetRequests 1.3.6.1.2.1.11.25
snmpOutGetNexsts 1.3.6.1.2.1.11.26
snmpOutSetRequests 1.3.6.1.2.1.11.27
snmpOutGetResponses 1.3.6.1.2.1.11.28
snmpOutTraps 1.3.6.1.2.1.11.29
snmpEnableAuthenTraps 1.3.6.1.2.1.11.30
snmpSilentDrops 1.3.6.1.2.1.11.31
snmpProxyDrops 1.3.6.1.2.1.11.32
```

## #SNMP modules

```
snmpSetSerialNo 1.3.6.1.6.3.1.1.6.1
snmpEngineID 1.3.6.1.6.3.10.2.1.1
snmpEngineBoots 1.3.6.1.6.3.10.2.1.2
snmpEngineTime 1.3.6.1.6.3.10.2.1.3
snmpEngineMaxMessageSize 1.3.6.1.6.3.10.2.1.4
snmpUnknownSecurityModels 1.3.6.1.6.3.11.2.1.1
snmpInvalidMsgs 1.3.6.1.6.3.11.2.1.2
snmpUnknownPDUHandlers 1.3.6.1.6.3.11.2.1.3
snmpTargetSpinLock 1.3.6.1.6.3.12.1.1
snmpUnknownContexts 1.3.6.1.6.3.12.1.5
usmStatsUnsupportedSecLevels 1.3.6.1.6.3.15.1.1.1
usmStatsNotInTimeWindows 1.3.6.1.6.3.15.1.1.2
usmStatsUnknownUserNames 1.3.6.1.6.3.15.1.1.3
usmStatsUnknownEngineIDs 1.3.6.1.6.3.15.1.1.4
usmStatsWrongDigests 1.3.6.1.6.3.15.1.1.5
usmStatsDecryptionErrors 1.3.6.1.6.3.15.1.1.6
usmUserSpinLock 1.3.6.1.6.3.15.1.2.1
usmUserTable 1.3.6.1.6.3.15.1.2.2
vacmContextTable 1.3.6.1.6.3.16.1.1
vacmSecurityToGroupTable 1.3.6.1.6.3.16.1.2
vacmAccessTable 1.3.6.1.6.3.16.1.4
vacmViewSpinLock 1.3.6.1.6.3.16.1.5.1
vacmViewTreeFamilyTable 1.3.6.1.6.3.16.1.5.2
```

## #SYSTEM

```
sysDescr 1.3.6.1.2.1.1.1
sysObjectID 1.3.6.1.2.1.1.2
sysUpTime 1.3.6.1.2.1.1.3
sysContact 1.3.6.1.2.1.1.4
sysName 1.3.6.1.2.1.1.5
sysLocation 1.3.6.1.2.1.1.6
```

```
sysServices 1.3.6.1.2.1.1.7
sysORLastChange 1.3.6.1.2.1.1.8
sysORTable 1.3.6.1.2.1.1.9
```

## #TCP-MIB

```
tcpRtoAlgorithm 1.3.6.1.2.1.6.1
tcpRtoMin 1.3.6.1.2.1.6.2
tcpRtoMax 1.3.6.1.2.1.6.3
tcpMaxConn 1.3.6.1.2.1.6.4
tcpActiveOpens 1.3.6.1.2.1.6.5
tcpPassiveOpens 1.3.6.1.2.1.6.6
tcpAttemptFails 1.3.6.1.2.1.6.7
tcpEstabResets 1.3.6.1.2.1.6.8
tcpCurrEstab 1.3.6.1.2.1.6.9
tcpInSegs 1.3.6.1.2.1.6.10
tcpOutSegs 1.3.6.1.2.1.6.11
tcpRetransSegs 1.3.6.1.2.1.6.12
tcpConnTable 1.3.6.1.2.1.6.13
tcpInErrs 1.3.6.1.2.1.6.14
tcpOutRsts 1.3.6.1.2.1.6.15
tcpConnectionTable 1.3.6.1.2.1.6.19
tcpListenerTable 1.3.6.1.2.1.6.20
```

## #UDP-MIB

```
udpInDatagrams 1.3.6.1.2.1.7.1
udpNoPorts 1.3.6.1.2.1.7.2
udpInErrors 1.3.6.1.2.1.7.3
udpOutDatagrams 1.3.6.1.2.1.7.4
udpTable 1.3.6.1.2.1.7.5
udpEndpointTable 1.3.6.1.2.1.7.7
dot3StatsTable 1.3.6.1.2.1.10.7.2
```