AOS-CX 10.06 IP Services Guide

8320, 8325, 8360 Switch Series

a Hewlett Packard
Enterprise company

Part Number: 5200-7705
Published: November 2020
Edition: 1

Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

This document describes features of the AOS-CX network operating system. It is intended for administrators responsible for installing, configuring, and managing Aruba switches on a network.

Applicable products

This document applies to the following products:

- Aruba 8320 Switch Series (JL479A, JL579A, JL581A)
- Aruba 8325 Switch Series (JL624A, JL625A, JL626A, JL627A)
- Aruba 8360 Switch Series (JL700A, JL701A, JL702A, JL703A, JL706A, JL707A, JL708A, JL709A, JL710A, JL711A)

Latest version available online

Updates to this document can occur after initial publication. For the latest versions of product documentation, see the links provided in **Support and other resources**.

Command syntax notation conventions

Convention	Usage
`example-text`	Identifies commands and their options and operands, code examples, filenames, pathnames, and output displayed in a command window. Items that appear like the example text in the previous column are to be entered exactly as shown and are required unless enclosed in brackets (`[ ]`).
`example-text`	In code and screen examples, indicates text entered by a user.
Any of the following: • *`<example-text>`* • `<example-text>` • *`example-text`* • *example-text*	Identifies a placeholder—such as a parameter or a variable—that you must substitute with an actual value in a command or in code: • For output formats where italic text cannot be displayed, variables are enclosed in angle brackets (`< >`). Substitute the text—including the enclosing angle brackets—with an actual value. • For output formats where italic text can be displayed, variables might or might not be enclosed in angle brackets. Substitute the text including the enclosing angle brackets, if any, with an actual value.
|	Vertical bar. A logical `OR` that separates multiple items from which you can choose only one. Any spaces that are on either side of the vertical bar are included for readability and are not a required part of the command syntax.

Table Continued

Convention	Usage
{ }	Braces. Indicates that at least one of the enclosed items is required.
[ ]	Brackets. Indicates that the enclosed item or items are optional.
… or . . .	Ellipsis: • In code and screen examples, a vertical or horizontal ellipsis indicates an omission of information. • In syntax using brackets and braces, an ellipsis indicates items that can be repeated. When an item followed by ellipses is enclosed in brackets, zero or more items can be specified.

About the examples

Examples in this document are representative and might not match your particular switch or environment.

The slot and port numbers in this document are for illustration only and might be unavailable on your switch.

Understanding the CLI prompts

When illustrating the prompts in the command line interface (CLI), this document uses the generic term `switch`, instead of the host name of the switch. For example:

```
switch>
```

The CLI prompt indicates the current command context. For example:

`switch>`

Indicates the operator command context.

`switch#`

Indicates the manager command context.

`switch(`*`CONTEXT-NAME`*`)#`

Indicates the configuration context for a feature. For example:

```
switch(config-if)#
```

Identifies the `interface` context.

Variable information in CLI prompts

In certain configuration contexts, the prompt may include variable information. For example, when in the VLAN configuration context, a VLAN number appears in the prompt:

```
switch(config-vlan-100)#
```

When referring to this context, this document uses the syntax:

```
switch(config-vlan-<VLAN-ID>)#
```

Where *<VLAN-ID>* is a variable representing the VLAN number.

Identifying switch ports and interfaces

Physical ports on the switch and their corresponding logical software interfaces are identified using the format:

member/*slot*/*port*

On the 83xx Switch Series

- *member*: Always 1. VSF is not supported on this switch.
- *slot*: Line module number. Always 1.
- *port*: Physical number of a port on a line module

For example, the logical interface `1/1/4` in software is associated with physical port 4 in slot 1 on member 1.

NOTE: If using breakout cables, the port designation changes to x:y, where x is the physical port and y is the lane when split to 4 x 10G or 4 x 25G. For example, the logical interface 1/1/4:2 in software is associated with lane 2 on physical port 4 in slot 1 on member 1.

ICMP Router Discovery Protocol (IRDP), an extension of the ICMP, is independent of any routing protocol. It allows hosts to discover the IP addresses of neighboring routers that can act as default gateways to reach devices on other IP networks.

IRDP operation

IRDP uses the following types of ICMP messages:

- Router advertisement (RA): Sent by a router to advertise IP addresses (including the primary and secondary IP addresses) and preference.
- Router solicitation (RS): Sent by a host to request the IP addresses of routers on the subnet.

An interface with IRDP enabled periodically broadcasts or multicasts an RA message to advertise its IP addresses. A receiving host adds the IP addresses to its routing table, and selects the IP address with the highest preference as the default gateway.

When a host attached to the subnet starts up, the host multicasts an RS message to request immediate advertisements. If the host does not receive any advertisements, it retransmits the RS several times. If the host does not discover the IP addresses of neighboring routers because of network problems, the host can still discover them from periodic RAs.

IRDP allows hosts to discover neighboring routers, but it does not suggest the best route to a destination. If a host sends a packet to a router that is not the best next hop, the host will receive an ICMP redirect message from the router.

IP address preference

Every IP address advertised in RAs has a preference value. A larger preference value represents a higher preference. The IP address with the highest preference is selected as the default gateway address.

You can specify the preference for IP addresses to be advertised on a router interface.

An address with the minimum preference value (-2147483648) will not be used as a default gateway address.

Lifetime of an IP address

An RA contains a lifetime field that specifies the lifetime of advertised IP addresses. If the host does not receive a new RA for an IP address within the address lifetime, the host removes the route entry.

All the IP addresses advertised by an interface have the same lifetime.

Advertising interval

A router interface with IRDP enabled sends out RAs randomly between the minimum and maximum advertising intervals. This mechanism prevents the local link from being overloaded by a large number of RAs sent simultaneously from routers.

As a best practice, shorten the advertising interval on a link that suffers high packet loss rates

Destination address of RA

An RA uses either of the following destination IP addresses:

- Broadcast address 255.255.255.255.
- Multicast address 224.0.0.1, which identifies all hosts on the local link.

By default, the destination IP address of an RA is the multicast address. If all listening hosts in a local area network support IP multicast, specify 224.0.0.1 as the destination IP address.

Proxy-advertised IP addresses

By default, an interface advertises its primary and secondary IP addresses. You can specify IP addresses of other gateways for an interface to proxy-advertise.

VRF support

In IP-based computer networks, virtual routing and forwarding (VRF) is a technology that allows multiple instances of a routing table to co-exist within the same router at the same time. Because the routing instances are independent, the same or overlapping IP addresses can be used without conflicting with each other.

IRDP is VRF aware. As the router advertisements and solicit processing occurs on the interface, packet is through the interface and corresponding VRF.

VSX synchronization

IRDP supports VSX synchronization. For more information on using VSX, see the *Virtual Switching Extension (VSX) Guide* for your switch and software version

Configuring IRDP

Prerequisites
A layer 3 interface.

Procedure

1. Enable IRDP on an interface with the command **`ip irdp`**.
2. Set the maximum hold time with the command **`ip irdp holdtime`**.
3. Set the maximum router advertisement interval with the command **`ip irdp maxadvertinterval`**.
4. Set the minimum router advertisement interval with the command **`ip irdp minadvertinterval`**.
5. Set the IRDP preference level with the command **`ip irdp preference`**.
6. Review IRDP configuration settings with the command **`show ip irdp`**.

Example

This example creates the following configuration:

- Enables IRDP on the layer 3 interface **`1/1/1`** with packet type set to **`broadcast`**.
- Sets the hold time to **`5000`** seconds.
- Sets the advertisement interval to **`30`** seconds.

- Sets the minimum advertisement interval to **25** seconds.
- Sets the IRDP preference level to **25**.

```
switch(config)# interface 1/1/1
switch(config-if)# ip irdp broadcast
switch(config-if)# ip irdp holdtime 5000
switch(config-if)# ip irdp maxadvertinterval 30
switch(config-if)# ip irdp minadvertinterval 25
switch(config-if)# ip irdp preference 25
```

IRDP commands

diag-dump irdp basic

Syntax

```
diag-dump irdp basic
```

Description

Displays diagnostic information for IRDP.

Command context

Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# diag-dump irdp basic
================================================================================
[Start] Feature irdp Time : Thu Jun  8 09:50:28 2017

================================================================================
--------------------------------------------------------------------------------
[Start] Daemon hpe-rdiscd
--------------------------------------------------------------------------------
Interface: 1/1/1 (state : Up)
rdisc ipv4 (enabled: 0, max:600, min:450, hold:1800, pref:0, isBcast:0)
Router IPs - 192.168.1.2,
Interface: 1/1/2 (state : Up)
rdisc ipv4 (enabled: 0, max:600, min:450, hold:1800, pref:0, isBcast:0)
Router IPs - 192.168.2.2,

--------------------------------------------------------------------------------
[End] Daemon hpe-rdiscd
--------------------------------------------------------------------------------
================================================================================
[End] Feature irdp
================================================================================
Diagnostic dump captured for feature irdp
```

ip irdp

Syntax

```
ip irdp [broadcast | multicast]

no ip irdp
```

Description

Enables IRDP on an interface and specifies the packet type that is used to send advertisements. By default, the packet type is set to `multicast`. IRDP is only supported on layer 3 interfaces.

The `no` form of this command disables IRDP on an interface.

Command context

`config`-if

Parameters

`broadcast`

Advertisements are sent as broadcast packets to IP address 255.255.255.255.

`multicast`

Advertisements are sent as multicast packets to the multicast group with IP address 24.0.0.1. Default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling IRDP on interface **`1/1/1`** with packet type set to the default value (multicast).

```
switch(config)# interface 1/1/1
switch(config-if)# ip irdp
```

Enabling IRDP on interface **`1/1/1`** with packet type set to **`broadcast`**.

```
switch(config)# interface 1/1/1
switch(config-if)# ip irdp broadcast
```

Disabling IRDP.

```
switch(config)# interface 1/1/1
switch(config-if)# no ip irdp
```

ip irdp holdtime

Syntax

```
ip irdp holdtime <TIME>
```

Description

Specifies the maximum amount of time the host will consider an advertisement to be valid until a newer advertisement arrives. When a new advertisement arrives, hold time is reset. Hold time must be greater than or equal to the maximum advertisement interval. Therefore, if the hold time for an advertisement expires, the host can reasonably conclude that the router interface that sent the advertisement is no longer available. The default hold time is three times the maximum advertisement interval.

Command context

`config-if`

Parameters

<TIME>

Specifies the lifetime of router advertisements sent from this interface. Range: 4 to 9000 seconds. Default: 1800 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Example

Setting the hold time for interface **1/1/1** to **5000** seconds:

```
switch(config)# interface 1/1/1
switch(config-if)# ip irdp holdtime 5000
```

ip irdp maxadvertinterval

Syntax

`ip irdp maxadvertinterval <TIME>`

Description

Specifies the maximum router advertisement interval.

Command context

`config`-if

Parameters

<TIME>

Specifies the maximum time allowed between the sending of unsolicited router advertisements. Range: 4 to 1800 seconds. Default: 600 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Example

Setting the advertisement interval for interface 1/1/1 to 30 seconds:

```
switch(config)# interface 1/1/1
switch(config-if)# ip irdp maxadvertinterval 30
```

ip irdp minadvertinterval

Syntax

`ip irdp minadvertinterval <TIME>`

Description

Specifies the minimum amount of time the switch waits between sending router advertisements. By default, this value is automatically set by the switch to be 75% of the value configured for maximum router advertisement interval. Use this command to override the automatically configured value.

Command context

`config-if`

Parameters

<TIME>

Specifies the minimum time allowed between the sending of unsolicited router advertisements. Range: 3 to 1800 seconds. Default: 450 seconds (75% of the default value for maximum router advertisement interval).

Authority

Administrators or local user group members with execution rights for this command.

Example

Setting the minimum advertisement interval for interface 1/1/1 to 25 seconds:

```
switch(config)# interface 1/1/1
switch(config-if)# ip irdp minadvertinterval 25
```

ip irdp preference

Syntax

`ip irdp preference <LEVEL>`

Description

Specifies the IRDP preference level. If a host receives multiple router advertisement messages from different routers, the host selects the router that sent the message with the highest preference as the default gateway.

Command context

`config-if`

Parameters

<LEVEL>

Specifies the IRDP preference level. Range: -2147483648 to 2147483647. Default: 0.

Authority

Administrators or local user group members with execution rights for this command.

Example

Setting the IRDP preference level for interface 1/1/1 to 25.

```
switch(config)# interface 1/1/1
switch(config-if)# ip irdp preference 25
```

show ip irdp

Syntax

```
show ip irdp [vsx-peer]
```

Description

Displays IRDP configuration settings.

Command context

Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Administrators or local user group members with execution rights for this command.

Example

```
switch# show ip irdp

ICMP Router Discovery Protocol

 Interface Status   Advertising Minimum  Maximum  Holdtime Preference
                    Address     Interval Interval
 --------- -------- ----------- -------- -------- -------- ----------
 1/1/1     Enabled  multicast   6        8        10       10
 1/1/2     Disabled multicast   450      600      1800     0
 1/1/3     Enabled  broadcast   450      600      1800     115
```

IPV6 RA provides a method for local IPV6 hosts to automatically configure their own IP address (and other settings such as a preferred DNS server) based on information advertised by switches/routers operating on the network.

IPv6 flags

Behavior of IPv6 hosts to IPv6 RA messages is controlled by the managed address configuration flag (M flag), and other stateful configuration flag (O flag).

M flag	O flag	Description
0	0	Indicates that no information is available via DHCPv6.
0	1	Indicates that other configuration information is available via DHCPv6. Examples of such information are DNS-related information or information on other servers within the network.
1	0	Indicates that addresses are available via Dynamic Host Configuration Protocol (DHCPv6).
1	1	If the M flag is set, the O flag is redundant and can be ignored because DHCPv6 will return all available configuration information.

Configuring IPv6 RA

Procedure

1. Enable transmission of IPv6 router advertisements with the command `no ipv6 nd suppress-ra`.
2. Optionally, configure IPv6 unicast address prefixes with the command `ipv6 nd prefix`.
3. Optionally, configure support for DNS name resolution with the commands `ipv6 nd ra dns server` and `ipv6 nd ra dns search-list`.
4. For most deployments, the default values for the following features do not need to be changed. If your deployment requires different settings, change the default values with the indicated command:

IPv6 RA setting	Default value	Command to change it
Number of neighbor solicitations to be sent when performing DAD.	1	ipv6 nd dad attempts
Number of neighbor entries in the ND cache.	131072	`ipv6 nd cache-limit`
Hop limit to be sent in the RA messages.	64	`ipv6 nd hop-limit`
MTU value to be sent in the RA messages.	1500 bytes	`ipv6 nd mtu`
Neighbor solicitation interval	1000 milliseconds	`ipv6 nd ns-interval`

Table Continued

IPv6 RA setting	Default value	Command to change it
Lifetime of a default router.	1800 seconds	`ipv6 nd ra lifetime`
Retrieval of an IPv6 address by devices.	Disabled	`ipv6 nd ra managed-config-flag`
Maximum interval between transmissions of IPv6 RAs.	600 seconds	`ipv6 nd ra max-interval`
Minimum interval between transmissions of IPv6 RAs.	200 seconds	`ipv6 nd ra min-interval`
Time that an interface considers a device to be reachable.	0 milliseconds (no limit)	`ipv6 nd ra reachable-time`
Retry period between ND solicitations.	0 (Use locally configured NS-interval)	`ipv6 nd ra retrans-timer`
Default routing preference for an interface.	Medium	`ipv6 nd router-preference`

5. Review IPv6 RA configuration settings with the commands `show ipv6 nd interface`, `show ipv6 nd interface prefix`, `show ipv6 nd ra dns server`, and `show ipv6 nd ra dns search-list`.

Example

This example creates the following configuration:

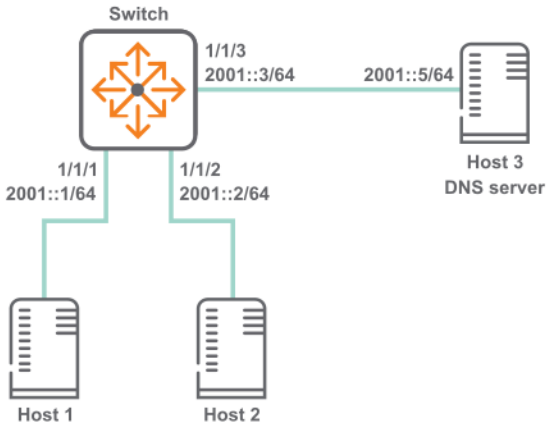
- Enables IPV6 RA on interface **1/1/3**.
- Sets the recursive DNS server address to **4001::1** with a lifetime of **400** seconds.
- Sets the minimum interval between transmissions to **3** seconds.
- Sets the maximum interval between transmissions to **13** seconds.
- Sets the lifetime of a default router to **1900** seconds.

```
switch(config)# interface 1/1/3
switch(config)# no ipv6 nd suppress-ra
switch(config-if)# ipv6 nd ra dns server 4001::1 lifetime 400
switch(config-if)# ipv6 nd ra min-interval 3
switch(config-if)# ipv6 nd ra max-interval 13
switch(config-if)# ipv6 nd ra lifetime 1900
switch(config-if)# end
switch# show ipv6 nd interface 1/1/3
Interface 1/1/3 is up
  Admin state is up
  IPv6 address:
    2006::1/64 [VALID]
  IPv6 link-local address: fe80::98f2:b321:368:6dc6/64 [VALID]
  ICMPv6 active timers:
      Last Router-Advertisement sent: 0 Secs
      Next Router-Advertisement sent in: 13 Secs
  Router-Advertisement parameters:
      Periodic interval: 3 to 13 secs
      Router Preference: medium
      Send "Managed Address Configuration" flag: false
      Send "Other Stateful Configuration" flag: false
```

```
      Send "Current Hop Limit" field: 64
      Send "MTU" option value: 1500
      Send "Router Lifetime" field: 1900
      Send "Reachable Time" field: 0
      Send "Retrans Timer" field: 0
      Suppress RA: false
      Suppress MTU in RA: true
  ICMPv6 error message parameters:
      Send redirects: false
  ICMPv6 DAD parameters:
      Current DAD attempt: 1
switch# show ipv6 nd ra dns server
Recursive DNS Server List on: 1/1/3
     Suppress DNS Server List: No
     DNS Server 1: 2001::1    lifetime 400
```

IPv6 RA scenario

In this scenario, two host computers are auto-configured with IP addresses using IPv6 RA. In addition, the switch provides the hosts with an address of a recursive DNS server. The physical topology of the network looks like this:



Procedure

1. Configure the interfaces with IPv6 addresses.

```
switch# config
switch(config)# interface 1/1/1
switch(config-if)# ipv6 address 2001::1/64
switch(config)# interface 1/1/2
switch(config-if)# ipv6 address 2001::2/64
switch(config)# interface 1/1/3
switch(config-if)# ipv6 address 2001::3/64
```

2. Enable transmission of all IPv6 RA messages.

```
switch(config-if)# no ipv6 nd suppress-ra
```

IPv6 RA commands

ipv6 address <global-unicast-address>

Syntax

```
ipv6 address <global-unicast-address>

no ipv6 address <global-unicast-address>
```

Description

Sets a global unicast address on the interface.

The `no` form of this command removes the global unicast address on the interface.

NOTE: This command automatically creates an IPv6 link-local address on the interface. However, it does not add the **ipv6 address link-local** command to the running configuration. If you remove the IPv6 address, the link-local address is also removed. To maintain the link-local address, you must manually execute the **ipv6 address link-local** command.

Command context

```
config-if
```

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Enabling a global unicast address:

```
switch(config)# interface 1/1/1
switch(config-if)#  ipv6 address 3731:54:65fe:2::a7
```

Disabling a global unicast address:

```
switch(config)# interface 1/1/1
switch(config-if)#  no ipv6 address 3731:54:65fe:2::a7
```

ipv6 address autoconfig

Syntax

```
ipv6 address autoconfig

no ipv6 address autoconfig
```

Description

Enables the interface to automatically obtain an IPv6 address using router advertisement information and the EUI-64 identifier.

The `no` form of this command disables address auto-configuration.

NOTE:

- A maximum of 15 autoconfigured addresses are supported.
- This command automatically creates an IPv6 link-local address on the interface. However, it does not add the **`ipv6 address link-local`** command to the running configuration. If you remove the IPv6 address, the link-local address is also removed. To maintain the link-local address, you must manually execute the **`ipv6 address link-local`** command.

Command context

`config-if`

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Usage

The IPv6 SLAAC feature lets the router obtain the IPv6 address for the interface it is configured through the SLAAC method. This feature is not available on the `mgmt` VRF.

Example

Enabling unicast autoconfiguring:

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 address autoconfig
```

Disabling unicast autoconfiguring:

```
switch(config)# interface 1/1/1
switch(config-if)# no ipv6 address autoconfig
```

ipv6 address link-local

Syntax

`ipv6 address link-local [<IPV6-ADDR>/<MASK>]`

Description

Enables IPv6 on the current interface. If no address is specified, an IPv6 link-local address is auto-generated for the interface. If an address is specified, auto-configuration is disabled and the specified address/mask is assigned to the interface.

To disable IPv6 link-local on the interface, remove `ipv6 address link-local`, `ipv6 address <global-ipv6-address>`, and `ipv6 address autoconfig` from the interface.

NOTE: This feature is not available on the management VRF.

Command context

`config-if`

Parameters

<IPV6-ADDR>

Specifies the IP address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F. You can use two colons (::) to represent consecutive zeros (but only once), remove leading zeros, and collapse a hextet of four zeros to a single 0. For example, this address `2222:0000:3333:0000:0000:0000:4444:0055` becomes `2222:0:3333::4444:55`.

<MASK>

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 128.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

Enabling IPv6 link-local on the interface:

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 address link-local
```

ipv6 nd cache-limit

Syntax

```
ipv6 nd cache-limit <CACHELIMIT>

no ipv6 nd cache-limit [<CACHELIMIT>]
```

Description

Configures the limit on the number of neighbor entries in the ND cache.

The `no` form of this command sets the cache limit to the default value.

Command context

```
config
```

Parameters

<CACHELIMIT>

Specifies the neighbor cache entries limit. Range: 1-131072. Default: 131072.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the cache limit to 20.

```
switch(config)# ipv6 nd cache-limit 20
```

ipv6 nd dad attempts

Syntax

```
ipv6 nd dad attempts <NUM-ATTEMPTS>

no ipv6 nd dad attempts [<NUM-ATTEMPTS>]
```

Description

Configures the number of neighbor solicitations to be sent when performing duplicate address detection (DAD) for a unicast address configured on an interface.

The `no` form of this command sets the number of attempts to the default value.

Command context

`config-if`

Parameters

`dad attempts <NUM-ATTEMPTS>`

Specifies the number of neighbor solicitations to send. Range: 0-15. Default: 1.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd dad attempts 5
```

ipv6 nd hop-limit

Syntax

```
ipv6 nd hop-limit <HOPLIMIT>

no ipv6 nd hop-limit [<HOPLIMIT>]
```

Description

Configures the hop limit to be sent in RAs.

The `no` form of this command resets the hop limit to 0. This reset eliminates the hop limit from the RAs that originate on the interface, so the host determines the hop limit.

Command context

`config-if`

Parameters

`hop-limit <HOPLIMIT>`

Specifies the hop limit. Range: 0-255. Default: 64.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd hop-limit 64
```

ipv6 nd mtu

Syntax

```
ipv6 nd mtu <MTU-VALUE>

no ipv6 nd mtu [<MTU-VALUE>]
```

Description

Configures the MTU size to be sent in the RA messages.

The `no` form of this command sets hop limit to the default value.

Command context

`config-if`

Parameters

<MTU-VALUE>

Specifies the MTU size. Range: 1280-65535 bytes. Default: 1500 bytes.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd mtu 1300
```

ipv6 nd ns-interval

Syntax

```
ipv6 nd ns-interval <TIME>

no ipv6 nd ns-interval [<TIME>]
```

Description

Configures the ND time between DAD neighbor solicitations sent for an unresolved destination, or between duplicate address detection neighbor solicitation requests. Increase this setting when neighbor solicitation retries or failures are occurring, or in a slow (WAN) network.

The `no` form of this command sets the ns-interval to the default value.

Command context

`config-if`

Parameters

<TIME>

Specifies the neighbor solicitation interval. Range: 1000-3600000 milliseconds. Default: 1000 milliseconds.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ns-interval 1200
```

ipv6 nd prefix

Syntax

```
ipv6 nd prefix <IPV6-ADDR>/<PREFIX-LEN>
     [no-advertise | [valid <LIFETIME-VALUE> preferred
     <LIFETIME-VALUE>] | no-autoconfig | no-onlink]

no ipv6 nd prefix <IPV6-ADDR>/<PREFIX-LEN> [no-advertise
     | [valid <LIFETIME-VALUE> preferred <LIFETIME-VALUE>
     ] | no-autoconfig | no-onlink]

ipv6 nd prefix default [no-advertise | [valid <LIFETIME-VALUE>
     preferred <LIFETIME-VALUE>] | no-autoconfig | no-onlink]}

no ipv6 nd prefix default [no-advertise | [valid <LIFETIME-VALUE>
     preferred <LIFETIME-VALUE>] | no-autoconfig | no-onlink]}
```

Description

Specifies prefixes for the routing switch to include in RAs transmitted on the interface. IPv6 hosts use the prefixes in RAs to autoconfigure themselves with global unicast addresses. The autoconfigured address of a host is composed of the advertised prefix and the interface identifier in the current link-local address of the host.

By default, advertise, autoconfig, and onlink are set.

The `no` form of this command removes the configuration on the interface.

Command context

`config-if`

Parameters

<IPV6-ADDR>/<PREFIX-LEN>

Specifies the IPv6 prefix to advertise in RA. Format: X:X::X:X/M

`default`

Specifies apply configuration to all on-link prefixes that are not individually set by the `ipv6 ra prefix` *<IPV6-ADDR>/<PREFIX-LEN>* command. It applies the same valid and preferred lifetimes, link state, autoconfiguration state, and advertise options to the advertisements sent for all on-link prefixes that are not individually configured with a unique lifetime. This also applies to the prefixes for any global unicast addresses configured later on the same interface.

Using default once, and then using it again with any new parameter values results in the new values replacing the former values in advertisements. If default is used without the `no-advertise`, `no-autoconfig`, or `no-onlink` parameter, the advertisement setting for the absent parameter is returned to its default setting.

`no-advertise`

Specifies do not advertise prefix in RA.

`valid <LIFETIME-VALUE>`

Specifies the total time, in seconds, the prefix remains available before becoming unusable. After preferred-lifetime expiration, any autoconfigured address is deprecated and used only for transactions only before preferred-lifetime expires. If the valid lifetime expires, the address becomes invalid.

You can enter a value in seconds or enter `valid infinite` which sets infinite lifetime. Default: 2,592,000 seconds which is 30 days. Range: 0–4294967294 seconds.

`preferred <LIFETIME-VALUE>`

Specifies the span of time during which the address can be freely used as a source and destination for traffic. This setting must be less than or equal to the corresponding valid–lifetime setting.

You can enter a value in seconds or enter `preferred infinite` which sets infinite lifetime. Default: 604,800 seconds which is seven days. Range: 0–4294967294 seconds.

`no-autoconfig`

Specifies do not use prefix for autoconfiguration.

`no-onlink`

Specifies do not use prefix for onlink determination.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd prefix 4001::1/64 valid 30 preferred 10 no-autoconfig no-onlink
```

ipv6 nd ra dns search-list

Syntax

```
ipv6 nd ra dns search-list <DOMAIN-NAME>  [lifetime <TIME>]

no ipv6 nd ra dns search-list <DOMAIN-NAME>
```

Description

Configures the DNS Search List (DNSSL) to include in Router Advertisements (RAs) transmitted on the interface.

The `no` form of this command removes the DNS Search List from the RAs transmitted on the interface.

Command context

`config-if`

Parameters

<DOMAIN-NAME>

Specifies the domain names for DNS queries.

`lifetime` ***<TIME>***

Specifies lifetime in seconds. Range: 4-4294967295 seconds. Default: 1800 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- DNSSL contains the domain names of DNS suffixes or IPv6 hosts to append to short, unqualified domain names for DNS queries.
- Multiple DNS domain names can be added to the DNSSL by using the command repeatedly.
- A maximum of eight server addresses are allowed.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra dns search-list test.com lifetime 500
```

ipv6 nd ra dns server

Syntax

`ipv6 nd ra dns server` *<IPV6-ADDR>* `[lifetime` *<TIME>*`]`

`no ipv6 nd ra dns server` *<IPV6-ADDR>*

Description

Configures the IPv6 address of a preferred Recursive DNS Server (RDNSS) to be included in Router Advertisements (RAs) transmitted on the interface.

The `no` form of this command removes the configured DNS server from the RAs transmitted on the interface.

Command context

`config-if`

Parameters

<IPV6-ADDR>

Specifies the RDNSS address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F. You can use two colons (::) to represent consecutive zeros (but only once), remove leading zeros, and collapse a hextet of four zeros to a single 0. For example, this address `2222:0000:3333:0000:0000:0000:4444:0055` becomes `2222:0:3333::4444:55`.

`lifetime` ***<TIME>***

Specifies IPv6 DNS server lifetime in seconds. Range: 4-4294967295 seconds. Default: 1800 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Including RDNSS information in RAs provides DNS server configuration for connected IPv6 hosts without requiring DHCPv6.
- Multiple servers can be configured on the interface by using the command repeatedly.
- A maximum of eight server addresses are allowed.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra dns server 2001::1 lifetime 400
```

ipv6 nd ra lifetime

Syntax

```
ipv6 nd ra lifetime <TIME>

no ipv6 nd ra lifetime [<TIME>]
```

Description

Configures the lifetime, in seconds, for the routing switch to be used as a default router by hosts on the current interface.

The `no` form of this command sets lifetime to the default of 1800 seconds.

Command context

```
config-if
```

Parameters

<TIME>

Specifies lifetime in seconds of a default router. A setting of 0 for default router lifetime in an RA indicates that the routing switch is not a default router on the interface. Range: 0-9000 seconds. Default: 1800 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- A given host on an interface refreshes the default router lifetime for a specific router each time the host receives an RA from that router.
- A specific router ceases to be a default router candidate for a given host if the default router lifetime expires before the host is updated with a new RA from the router.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra lifetime 1200
```

ipv6 nd ra managed-config-flag

Syntax

```
ipv6 nd ra managed-config-flag

no ipv6 nd ra managed-config-flag
```

Description

Controls the M flag setting in RAs the router transmits on the current interface. Enable the M flag to indicate that hosts can obtain IP address through DHCPv6. The M flag is disabled by default.

The `no` form of this command turns off (disables) the M flag.

Command context

```
config-if
```

Authority

Administrators or local user group members with execution rights for this command.

Usage

- Enabling the M flag directs hosts to acquire their IPv6 addressing for the current interface from a DHCPv6 server.
- When the M-bit is enabled, receiving hosts ignore the O flag setting, which is configured using the command `ipv6 nd ra other-config-flag`.
- When the M-bit is disabled (the default), receiving hosts expect to receive their IPv6 addresses from RA.

M flag	O flag	Description
0	0	Indicates that no information is available via DHCPv6.
0	1	Indicates that other configuration information is available via DHCPv6. Examples of such information are DNS-related information or information on other servers within the network.
1	0	Indicates that addresses are available via Dynamic Host Configuration Protocol (DHCPv6).
1	1	If the M flag is set, the O flag is redundant and can be ignored because DHCPv6 will return all available configuration information.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra managed-config-flag
```

ipv6 nd ra max-interval

Syntax

```
ipv6 nd ra max-interval <TIME>

no ipv6 nd ra max-interval [<TIME>]
```

Description

Configures the maximum interval between transmissions of IPv6 RAs on the interface. The interval between RA transmissions on an interface is a random value that changes every time an RA is sent. The interval is calculated to be a value between the current max-interval and min-interval settings.

The `no` form of this command returns the setting to its default, provided the default value is less than the default lifetime value.

Command context

`config-if`

Parameters

<TIME>

Specifies the maximum advertisement time in seconds. Range: 4-1800. Default: 600 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- This value has one setting per interface. The setting does not apply to RAs sent in response to a router solicitation received from another device.
- Attempting to set max-interval to a value that is not sufficiently larger than the current min-interval also results in an error message.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra max-interval 30
```

ipv6 nd ra min-interval

Syntax

```
ipv6 nd ra min-interval <TIME>

no ipv6 nd ra min-interval [<TIME>]
```

Description

Configures the minimum interval between transmissions of IPv6 RAs on the interface. The interval between RA transmissions on an interface is a random value that changes every time an RA is sent. The interval is calculated to be a value between the current max-interval and min-interval settings.

The `no` form of this command returns the setting to its default, provided the default value is less than the current max-interval setting.

Command context

`config-if`

Parameters

<TIME>

Specifies a minimum advertisement time in seconds. Range: 3-1350. Default: 200 seconds.

Authority

Administrators or local user group members with execution rights for this command.

Usage

- This value has one setting per interface and does not apply to RAs sent in response to a router solicitation received from another device.
- The min-interval must be less than the max-interval. Attempting to set min-interval to a higher value results in an error message.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra  min-interval 25
```

ipv6 nd ra other-config-flag

Syntax

```
ipv6 nd ra other-config-flag

no ipv6 nd ra other-config-flag
```

Description

Controls the O-bit in RAs the router transmits on the current interface; but is ignored unless the M-bit is disabled in RAs. Configure to set the O-bit in RA messages for host to obtain network parameters through DHCPv6. The other-config-flag is disabled by default.

For more information on configuring the M-bit, see `ipv6 nd ra managed-config-flag`.

The `no` form of this command turns off (disables) the setting for this command in RAs.

Command context

`config-if`

Authority

Administrators or local user group members with execution rights for this command.

Usage

Enabling the O-bit while the M-bit is disabled directs hosts on the interface to acquire their other configuration information from DHCPv6. Examples of such information are DNS-related information or information on other servers within the network.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra other-config-flag
```

ipv6 nd ra reachable-time

Syntax

```
ipv6 nd ra reachable-time <TIME>

no ipv6 nd ra reachable-time [<TIME>]
```

Description

Sets the amount of time that the interface considers a device to be reachable after receiving a reachability confirmation from the device.

The `no` form of this command sets the reachable time to the default value of 0. (no limit).

Command context

```
config-if
```

Parameters

<TIME>

Specifies the reachable time in milliseconds. Range: 1000-3600000. Default: 0 (no limit).

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra reachable-time 2000
```

ipv6 nd ra retrans-timer

Syntax

```
ipv6 nd ra retrans-timer <TIME>

no ipv6 nd ra retrans-timer [<TIME>]
```

Description

Configures the period (retransmit timer) between ND solicitations sent by a host for an unresolved destination, or between DAD neighbor solicitation requests. By default, hosts on the interface use their own locally configured NS-interval settings instead of using the value received in the RAs.

Increase this timer when neighbor solicitation retries or failures are occur, or in a "slow" (WAN) network.

The `no` form of this command sets the value to the default of 0.

Command context

```
config-if
```

Parameters

<TIME>

Specifies the retransmit timer value in milliseconds. Range: 0 - 4294967295 milliseconds. Default: 0 (Use locally configured NS-interval).

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra retrans-timer 400
```

ipv6 nd router-preference

Syntax

```
ipv6 nd router-preference {high | medium | low}

no ipv6 nd router-preference [high | medium | low]
```

Description

Specifies the value that is set in the Default Router Preference (DRP) field of Router Advertisements (RAs) that the switch sends from an interface. An interface with a DRP value of high will be preferred by other devices on the network over interfaces with an RA value of medium or low.

The `no` form of this command set the value to the default of medium.

Command context

```
config-if
```

Parameters

`high`

Sets DRP to high.

`medium`

Sets DRP to medium. Default.

`low`

Sets DRP to low.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd router-preference high
```

ipv6 nd suppress-ra

Syntax

```
ipv6 nd suppress-ra [<SUPPRESS-OPTION>]

no ipv6 nd ra supress-ra [<SUPPRESS-OPTION>]
```

Description

Configures suppression of IPv6 Router Advertisement transmissions on an interface.

The `no` form of this command restores transmission of IPv6 Router Advertisement and options.

Command context

`config-if`

Parameters

`suppress-ra [<SUPPRESS-OPTION>]`

Specifies suppressing RA transmissions. Entering `suppress-ra` without any options, suppresses all RA messages (default). Or you can enter one of the following options.

`dnssl`

Specifies suppressing DNSSL options in RA messages.

`mtu`

Specifies suppressing MTU options in RA messages.

`rdnss`

Specifies suppressing RDNSS options in RA messages.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd suppress-ra mtu dnssl rdnss
switch(config-if)# no ipv6 nd suppress-ra mtu dnssl rdnss
```

show ipv6 nd global traffic

Syntax

```
show ipv6 nd global traffic [vsx-peer]
```

Description

Displays IPV6 Neighbor Discovery traffic details on a device.

Command context

Operator (>) or Manager (#)

Parameters

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show ipv6 nd global traffic
  ICMPv6 packet Statistics (sent/received)
    Total Messages                :        18/0
    Error Messages                :        0/0
    Destination Unreachables      :        0/0
    Time Exceeded                 :        0/0
    Parameter Problems            :        0/0
    Echo Request                  :        0/0
    Echo Replies                  :        0/0
    Redirects                     :        0/0
    Packet Too Big                :        0/0
    Router Advertisements         :        4/0
    Router Solicitations          :        0/0
    Neighbor Advertisements       :        0/0
    Neighbor Solicitations        :        3/0
    Duplicate router RA received :         0/0
  ICMPv6 MLD Statistics (sent/received)
    V1 Queries :           0/0
    V2 Queries :           0/0
    V1 Reports :           0/0
    V2 Reports :           11/0
    V1 Leaves  :           0/0
```

show ipv6 nd interface

Syntax

```
show ipv6 nd interface [<IF-NAME> | all-vrfs | vrf <VRF-NAME>]
     [vsx-peer]
```

Description

Displays neighbor discovery information for an interface. If no options are specified, displays information for the default VRF.

Command context

Operator (>) or Manager (#)

Parameters

<IF-NAME>

Displays information about the specified IPv6 enabled interface.

all-vrfs

Displays information about interfaces in all VRFs.

vrf *<VRF-NAME>*

Displays information about interfaces in a particular VRF. Or, if *<VRF-NAME>* is not specified, information for the default VRF is displayed.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing information for all VRFs:

```
switch# show ipv6 nd interface all-vrfs

List of IPv6 Interfaces for VRF default
Interface 1/1/1 is up
  Admin state is up
  IPv6 address:
  IPv6 link-local address: fe80::7272:cfff:fee7:a8b9/64 [VALID]
  ICMPv6 active timers:
      Last Router-Advertisement sent:
      Next Router-Advertisement sent in:
  Router-Advertisement parameters:
      Periodic interval: 200 to 600 secs
      Router Preference: medium
      Send "Managed Address Configuration" flag: false
      Send "Other Stateful Configuration" flag: false
      Send "Current Hop Limit" field: 64
      Send "MTU" option value: 1500
      Send "Router Lifetime" field: 1800
      Send "Reachable Time" field: 0
      Send "Retrans Timer" field: 0
      Suppress RA: true
      Suppress MTU in RA: true
  ICMPv6 error message parameters:
      Send redirects: false
  ICMPv6 DAD parameters:
      Current DAD attempt: 1


List of IPv6 Interfaces for VRF red
Interface 1/1/2 is up
  Admin state is up
  IPv6 address:
    2001::1/64 [VALID]
  IPv6 link-local address: fe80::7272:cfff:fee7:a8b9/64 [VALID]
  ICMPv6 active timers:
      Last Router-Advertisement sent:
      Next Router-Advertisement sent in:
  Router-Advertisement parameters:
      Periodic interval: 200 to 600 secs
      Router Preference: medium
      Send "Managed Address Configuration" flag: false
      Send "Other Stateful Configuration" flag: false
      Send "Current Hop Limit" field: 64
      Send "MTU" option value: 1500
      Send "Router Lifetime" field: 1800
      Send "Reachable Time" field: 0
      Send "Retrans Timer" field: 0
      Suppress RA: true
      Suppress MTU in RA: true
  ICMPv6 error message parameters:
      Send redirects: false
  ICMPv6 DAD parameters:
      Current DAD attempt: 1
```

Showing information for interface 1/1/1:

```
switch# show ipv6 nd interface 1/1/1
Interface 1/1/1 is up
  Admin state is up
  IPv6 address:
  IPv6 link-local address: fe80::7272:cfff:fee7:a8b9/64 [VALID]
  ICMPv6 active timers:
      Last Router-Advertisement sent:
      Next Router-Advertisement sent in:
  Router-Advertisement parameters:
      Periodic interval: 200 to 600 secs
      Router Preference: high
      Send "Managed Address Configuration" flag: false
      Send "Other Stateful Configuration" flag: false
      Send "Current Hop Limit" field: 64
      Send "MTU" option value: 1500
      Send "Router Lifetime" field: 1800
      Send "Reachable Time" field: 0
      Send "Retrans Timer" field: 0
      Suppress RA: true
      Suppress MTU in RA: true
  ICMPv6 error message parameters:
      Send redirects: false
  ICMPv6 DAD parameters:
      Current DAD attempt: 1
```

Showing information for the default VRF:

```
switch# show ipv6 nd interface

List of IPv6 Interfaces for VRF default
Interface 1/1/1 is up
  Admin state is up
  IPv6 address:
      2001::1/64  [VALID]
  IPv6 link-local address: fe80::7272:cfff:fee7:a8b9/64 [VALID]
  ICMPv6 active timers:
      Last Router-Advertisement sent: 6 Secs
      Next Router-Advertisement sent in: 7 Secs
  Router-Advertisement parameters:
      Periodic interval: 3 to 13 secs
      Router Preference: medium
      Send "Managed Address Configuration" flag: false
      Send "Other Stateful Configuration" flag: false
      Send "Current Hop Limit" field: 64
      Send "MTU" option value: 1500
      Send "Router Lifetime" field: 1900
      Send "Reachable Time" field: 0
      Send "Retrans Timer" field: 0
      Suppress RA: true
      Suppress MTU in RA: true
  ICMPv6 error message parameters:
      Send redirects: false
  ICMPv6 DAD parameters:
      Current DAD attempt: 1
```

show ipv6 nd interface prefix

Syntax

```
show ipv6 nd interface prefix [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows IPv6 prefix information for all VRFs or a specific VRF. If no options are specified, shows information for the default VRF.

Command context

Operator (>) or Manager (#)

Parameters

`all-vrfs`

Shows prefix information for all VRFs.

`vrf <VRF-NAME>`

Name of a VRF.

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing prefix information for the default VRF:

```
switch# show ipv6 nd interface prefix

List of IPv6 Interfaces for VRF default
List of IPv6 Prefix advertised on 1/1/1
   Prefix : 4545::/65
   Enabled : Yes
   Validlife time : 2592000
   Preferred lifetime : 604800
   On-link : Yes
   Autonomous : Yes
```

Showing information for VRF red:

```
switch# show ipv6 nd interface prefix vrf red

List of IPv6 Interfaces for VRF red
List of IPv6 Prefix advertised on 1/1/2
   Prefix : 2001::/64
   Enabled : Yes
   Validlife time : 2592000
   Preferred lifetime : 604800
   On-link : Yes
   Autonomous : Yes
```

show ipv6 nd ra dns search-list

Syntax

```
show ipv6 nd ra dns search-list [vsx-peer]
```

Description

Displays domain name information on all interfaces.

Command context

Operator (>) or Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra dns search-list test.com
switch# show ipv6 nd ra dns search-list
Recursive DNS Search List on: 1
     Suppress DNS Search List: Yes
     DNS Search 1: test.com    lifetime  1800
```

show ipv6 nd ra dns server

Syntax

```
show ipv6 nd ra dns server [vsx-peer]
```

Description

Displays DNS server information on all interfaces.

Command context

Operator (>) or Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 nd ra dns server 2001::1
switch# show ipv6 nd ra dns server
Recursive DNS Server List on: 1
```

```
    Suppress DNS Server List: Yes
    DNS Server 1: 2001::1    lifetime 1800
```

sFlow is a technology for monitoring traffic in switched or routed networks. The sFlow monitoring system is comprised of:

- An sFlow Agent that runs on a network device, such as a switch. The agent uses sampling techniques to capture information about the data traffic flowing through the device and forwards this information to an sFlow collector.
- An sFlow Collector that receives monitoring information from sFlow agents. The collector stores this information so that a network administrator can analyze it to understand network data flow patterns. One sFlow collector can recieve the data from many sFlow agents.

NOTE: The sFlow UDP datagrams sent to a collector are not encrypted, therefore any sensitive information contained in an sFlow sample is exposed.

sFlow agent

The sFlow agent on the switch provides ingress sampling of all forwarded layer 2 and layer 3 traffic on LAG and Ethernet ports. High-availability is supported (packet sampling continues to work after switch-over).

The sFlow agent can communicate with up to three sFlow collectors at the same time. The agent communicates with collectors only on the default VRF.

Although you can configure very high sampling rates, the switch may drop samples if it cannot handle the rate of sampled packets. High sampling rates may also cause high CPU usage resulting in control plane performance issues.

A single sFlow datagram sent to a collector contains multiple flow and counter samples. The total number of samples an sFlow datagram can contain varies depending on the settings for header size and maximum datagram size.

Default settings

- sFlow is disabled on all interfaces.
- Collector port: UDP port 6343.
- Sampling rate: 4096.
- Polling interval: 30 seconds.
- Header size: 128 bytes.
- Max datagram size: 1400 bytes.

Supported features

- Global sampling rate
- Interface counters polling
- Agent IP configuration for IPv4 and IPv6
- Header size configuration

- Max datagram size configuration
- Ingress sampling for all forwarded traffic (L2, L3)
- Enable/Disable sFlow per interface
- Support for three remote collectors
- An out-of-band collector can be defined on the management VRF
- A collector can be defined on the non-default VRF
- Sampling on Ethernet and LAG interfaces
- High availability support (sampling continues to work after switch-over)
- Source IP support (setting source IP for sFlow datagrams sent to a remote collector)

Limitations

- No sampling of egress traffic
- Sampling rate cannot be set per interface (global only)
- sFlow is not configurable via SNMP

Configuring the sFlow agent

Procedure

1. Configure one or more sFlow collectors with the command **`sflow collector`**. This determines where the sFlow agent sends sFlow information.
2. Enable the sFlow agent on all interfaces, or on a specific interface, with the command **`sflow`**.
3. Define the address of the sFlow agent with the command **`sflow agent-ip`**.
4. By default, the source IP address for sFlow datagrams is set to the IP address of the outgoing switch interface on which the sFlow client is communicating with a collector. Since the switch can have multiple routing interfaces, datagrams can potentially be sent on different paths at different times, resulting in different source IP addresses for the same client. To resolve this issue, define a single source IP address. For details, see *Single source IP address* in the *Fundamentals Guide*.
5. For most deployments, the default values for the following settings do not need to be changed. If your deployment requires different settings, change the default values with the indicated commands:

sFlow setting	Default value	Command to change it
Rate at which packets are sampled.	1 in every 4096 packets	`sflow sampling`
Rate at which the switch sends data to an sFlow collector.	30 seconds	`sflow polling`
Size of the sFlow header.	128 bytes	`sflow header-size`
Maximum size of an sFlow datagram.	1400 bytes	`sflow max-datagram-size`

6. Review sFlow configuration settings with the command **`show sflow`**.

Example

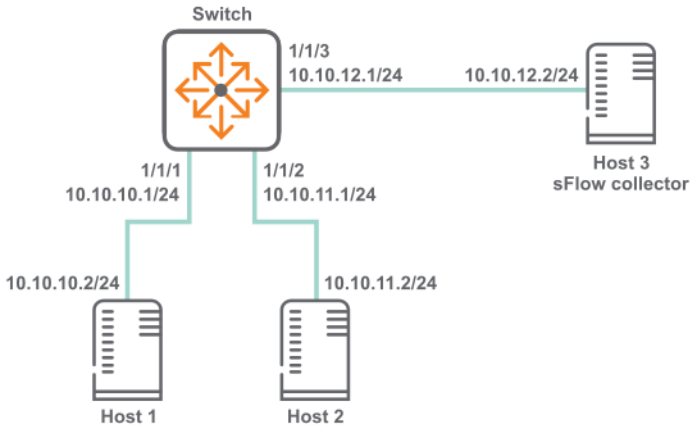
This example creates the following configuration:

- Configures an sFlow collector with the IP address **10.10.20.209** .
- Enables the sFlow agent on all interfaces.
- Defines the sFlow agent IP address to be **10.10.1.5**.

```
switch(config)# sflow collector 10.10.20.209
switch(config)# sflow
switch(config)# sflow agent-ip 10.0.0.1
```

sFlow scenario

In this scenario, two hosts send sFlow traffic through a switch to an sFlow collector. The physical topology of the network looks like this:



Procedure

1. Enable sFlow globally.

   ```
   switch# config
   switch(config)# sflow
   ```

2. Set the sFlow agent IP address to **10.10.12.1**.

   ```
   switch(config)# sflow agent-ip 10.10.12.1
   ```

3. Set the sFlow collector IP address to **10.10.12.2**.

   ```
   switch(config)# sflow collector 18.2.2.2
   ```

4. Configure sFLow sampling rate and polling interval.

   ```
   switch(config)# sflow sampling 5000
   switch(config)# sflow polling 20
   ```

5. Configure interface **1/1/1** with IP address **10.10.10.1/24.**

   ```
   switch(config)# interface 1/1/1
   switch(config-if)# no shutdown
   ```

```
switch(config-if)# ip address 10.10.10.1/24
switch(config)# quit
```

6. Configure interface **1/1/2** with IP address **10.10.11.1/24**.

```
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-if)# ip address 10.10.11.1/24
switch(config)# quit
```

7. Configure interface **1/1/3** with IP address **10.10.12.1/24**.

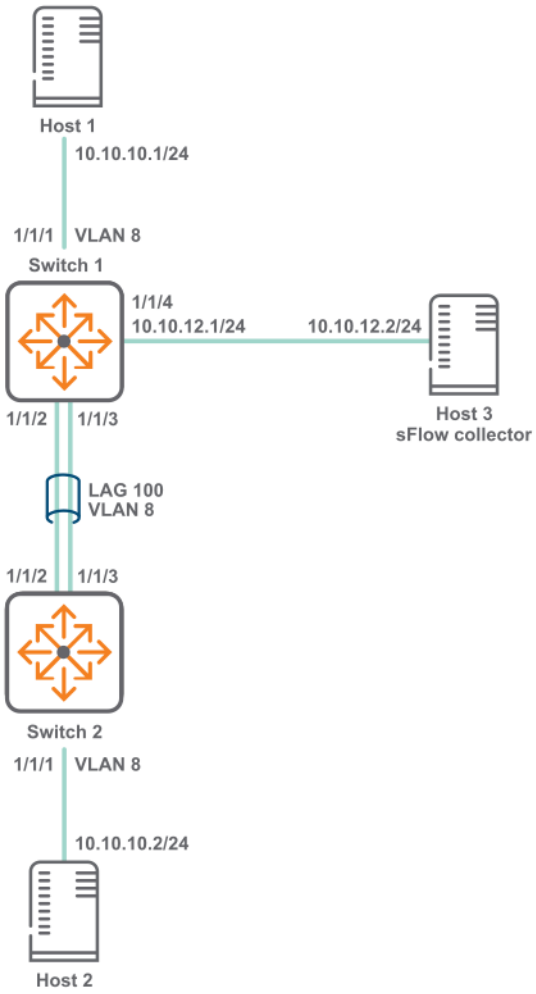
```
switch(config)# interface 1/1/3
switch(config-if)# no shutdown
switch(config-if)# ip address 10.10.12.1/24
switch(config)# quit
```

8. Verify sFlow configuration.

```
switch# show sflow

sFlow Global Configuration
----------------------------------------
sFlow                          enabled
Collector IP/Port/Vrf          10.10.12.2/6343/default
Agent Address                  10.10.12.1
Sampling Rate                  5000
Polling Interval               20
Header Size                    128
Max Datagram Size              1400

sFlow Status
----------------------------------------
Running - Yes

sFlow Statistics
----------------------------------------
Number of Samples              25
```

sFlow scenario 2

In this scenario, two hosts connected to different switches send sFlow traffic to a collector. A LAG is used to connect the two switches. The physical topology of the network looks like this:



Procedure

1. Configure switch 1.

   a. Enable sFlow globally.

   ```
   switch# config
   switch(config)# sflow
   ```

   b. Set the sFlow agent IP address to `10.10.12.1`.

   ```
   switch(config)# sflow agent-ip 10.10.12.1
   ```

   c. Set the sFlow collector IP address to `10.10.12.2`.

   ```
   switch(config)# sflow collector 10.10.12.2
   ```

   d. Configure sFLow sampling rate and polling interval.

   ```
   switch(config)# sflow sampling 5000
   switch(config)# sflow polling 10
   ```

   e. Create VLAN `8`.

   ```
   switch(config)# vlan 8
   switch(config-vlan-8)# no shutdown
   switch(config)# exit
   ```

f. Define LAG **100** and assign VLAN **vlan 8** to it.

```
switch(config)# interface lag 100
switch(config-lag-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-lag-if)# vlan access 8
switch(config-lag-if)# lacp mode active
```

g. Configure interface **1/1/1**.

```
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-if)# vlan access 8
```

h. Configure interface **1/1/2** and **1/1/3** as members of LAG **100.**

```
switch# (config)#interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# lag 100
switch(config-if)# exit
switch(config)-if# interface 1/1/3
switch(config-if)# no shutdown
switch(config-if)# lag 100
switch(config-if)# exit
```

i. Configure interface **1/1/4** with IP address **10.10.12.1/24**.

```
switch# (config)#interface 1/1/4
switch(config-if)# no shutdown
switch(config-if)# ip address 10.10.12.1/24
switch(config-if)# quit
```

j. Verify sFlow configuration.

```
switch# show sflow

sFlow Global Configuration
----------------------------------------
sFlow                          enabled
Collector IP/Port/Vrf          10.10.12.2/6343/default
Agent Address                  10.10.12.1
Sampling Rate                  5000
Polling Interval               10
Header Size                    128
Max Datagram Size              1400

sFlow Status
----------------------------------------
Running - Yes

sFlow Statistics
----------------------------------------
Number of Samples              120
```

2. Configure switch 2.

a. Create VLAN **8**.

```
switch(config)# vlan 8
switch(config-vlan-8)# no shutdown
switch(config)# exit
```

b. Define LAG **100** and assign VLAN **vlan 8** to it.

```
switch(config)# interface lag 100
switch(config-lag-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-lag-if)# vlan access 8
switch(config-lag-if)# lacp mode active
```

c. Configure interface **1/1/1**.

```
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
switch(config-lag-if)# no routing
switch(config-if)# vlan access 8
```

d. Configure interface **1/1/2** and **1/1/3** as members of LAG **100**.

```
switch# (config)#interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# lag 100
switch(config-if)# exit
switch(config)-if# interface 1/1/3
switch(config-if)# no shutdown
switch(config-if)# lag 100
switch(config-if)# exit
```

sFlow agent commands

sflow

Syntax

```
sflow

no sflow
```

Description

Enables the sFlow agent.

- In the `config` context, this command enables the sFlow agent globally on all interfaces.
- In an `config-if` context, this command enables the sFlow agent on a specific interface. sFlow cannot be enabled on a member of a LAG, only on the LAG.

The sFlow agent is disabled by default.

The `no` form of this command disables the sFlow agent and deletes all sFlow configuration settings, either globally, or for a specific interface.

Command context

```
config

config-if
```

Parameters

None.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling sFlow globally on all interfaces:

```
switch(config)# sflow
```

Disabling sFlow globally on all interfaces:

```
switch(config)# no sflow
```

Enabling sFlow on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# sflow
```

Disabling sFlow on interface **1/1/1**:

```
switch(config)# interface 1/1/1
switch(config-if)# no sflow
```

Enabling sFlow on interface **lag100**:

```
switch(config)# interface lag100
switch(config-if)# sflow
```

Disabling sFlow on interface **lag100**:

```
switch(config)# interface lag100
switch(config-if)# no sflow
```

sflow agent-ip

Syntax

```
sflow agent-ip <IP-ADDR>

no sflow agent-ip [<IP-ADDR>]
```

Description

Defines the IP address of the sFlow agent to use in sFlow datagrams. This address must be defined for sFlow to function. HPE recommends that the address:

- can uniquely identify the switch
- is reachable by the sFlow collector
- does not change with time

The `no` form of this command deletes the IP address of the sFlow agent. This causes sFlow to stop working and no datagrams will be sent to the sFlow collector.

Command context

`config`

Parameters

<IP-ADDR>

Specifies an IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F. The agent address is used to identify the switch in all sFlow datagrams sent to sFlow collectors. It is usually set to an IP address on the switch that is reachable from an sFlow collector.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the agent address to **`10.10.10.100`**:

```
switch(config)# sflow agent-ip 10.0.0.100
```

Setting the agent address to **`2001:0db8:85a3:0000:0000:8a2e:0370:7334`**:

```
switch(config)# sflow agent-ip 2001:0db8:85a3:0000:0000:8a2e:0370:7334
```

Removing the address configuration from the switch, which results in sFlow being disabled:

```
switch(config)# no sflow agent-ip
```

sflow collector

Syntax

```
sflow collector <IP-ADDR> [port <PORT>] [vrf <VRF>]

no sflow collector <IP-ADDR> [port <PORT>] [vrf <VRF>]
```

Description

Defines a collector to which the sFlow agent sends data. Up to three collectors can be defined. At least one collector should be defined, and it must be reachable from the switch for sFlow to work.

Command context

`config`

Parameters

`collector <IP-ADDR>`

Specifies the IP address of a collector in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

`port <PORT>`

Specifies the UDP port on which to send information to the sFlow collector. Range: 0 to 65536. Default: 6343.

vrf *<VRF>*

Specifies the VRF on which to send information to the sFlow collector. The VRF must be defined on the switch. If no VRF is specified, the default VRF (`default`) is used.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defining a collector with IP address **10.10.10.100** on UDP port **6400**:

```
switch(config)# sflow collector 10.0.0.1 port 6400
```

sflow disable

Syntax

```
sflow disable
```

Description

Disables the sFlow agent, but retains any existing sFlow configuration settings. The settings become active if the sFlow agent is re-enabled.

Command context

```
config
```

Parameters

None.

Authority

Administrators or local user group members with execution rights for this command.

Example

Disabling sFlow support:

```
switch(config)# sflow disable
```

sflow header-size

Syntax

```
sflow header-size <SIZE>

no sflow header-size [<SIZE>]
```

Description

Sets the sFlow header size in bytes.

The `no` form of this command sets the header size to the default value of 128.

Command context

```
config
```

Parameters

header-size *<SIZE>*

Specifies the sFlow header size in bytes. Range: 64 to 256. Default: 128.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the header size to **64** bytes:

```
switch(config)# sflow header-size 64
```

Setting the header size to the default value of **128** bytes:

```
switch(config)# no sflow header-size
```

sflow max-datagram-size

Syntax

```
sflow max-datagram-size <SIZE>

no sflow max-datagram-size [<SIZE>]
```

Description

Sets the maximum number of bytes that are sent in one sFlow datagram.

The `no` form of this command sets maximum number of bytes to the default value of 1400.

Command context

`config`

Parameters

max-datagram-size *<SIZE>*

Specifies the maximum datagram size in bytes. Range: 1 to 9000. Default: 1400.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the datagram size to **1000** bytes:

```
switch(config)# sflow max-datagram-size 1000
```

Setting the header size to the default value of **1400** bytes:

```
switch(config)# no sflow max-datagram-size
```

sflow polling

Syntax

```
sflow polling <INTERVAL>

no sflow polling [<INTERVAL>]
```

Description

Defines the global polling interval for sFlow in seconds.

The `no` form of this command sets the polling interval to the default value of 30 seconds.

Command context

```
config
```

Parameters

<INTERVAL>

Specifies the polling interval in seconds. Range: 10 to 3600. Default: 30.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the polling interval to 10:

```
switch(config)# sflow polling 10
```

Setting the polling interval to the default value.

```
switch(config)# no sflow polling
```

sflow sampling

Syntax

```
sflow sampling <RATE>

no sflow sampling [<RATE>]
```

Description

Defines the global sampling rate for sFlow in number of packets. The default sampling rate is 4096, which means that one in every 4096 packets is sampled. A warning message is displayed when the sampling rate is set to less than 4096 and proceeds only after user confirmation.

The `no` form of this command sets the sampling rate to the default value of 4096.

Command context

```
config
```

Parameters

sampling *<RATE>*

Specifies the sampling rate. Range: 1 to 1000000000. Default: 4096.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the sampling rate to **5000**:

```
switch(config)# sflow sampling 5000
```

Setting the sampling rate to the default:

```
switch(config)# no sflow sampling
```

Setting the sampling rate to **1000**:

```
switch(config)# sflow sampling 1000
Setting the sFlow sampling rate lower than 4096 is not recommended and might
affect system performance.
Do you want to continue [y/n]? y
switch(config)#
```

show sflow

Syntax

```
show sflow [interface <INTERFACE-NAME>] [vsx-peer]
```

Description

Shows sFlow configuration settings and statistics for all interfaces, or for a specific interface

Command context

Manager (#)

Parameters

interface *<INTERFACE-NAME>*

Specifies the name of an interface on the switch.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Showing sFlow information for all interfaces:

```
switch# show sflow
sFlow Global Configuration
------------------------------------------
sFlow                          enabled
Collector IP/Port/Vrf          10.0.0.2/6343/default
                               10.0.0.3/6400/default
Agent Address                  10.0.0.1
Sampling Rate                  1024
```

```
Polling Interval              30
Header Size                   128
Max Datagram Size             1400

sFlow Status
----------------------------------------
Running - Yes

sFlow Statistics
----------------------------------------
Number of Samples             200

- Agent address is not configured.
```

Showing sFlow information for interface **1/1/1**:

```
switch# show sflow 1/1/1
sFlow configuration - Interface 1
----------------------------------------
sFlow                         enabled
Sampling Rate                 1024
Number of Samples             30
```

The Dynamic Host Configuration Protocol (DHCP) enables the automatic assignment of IP addresses and other configuration settings to network devices.

DHCP is composed of three components: DHCP server, DHCP client, and DHCP relay agent.

The DHCP server contains the IP addresses and configuration settings for a network as defined by a network administrator. It responds to DHCP requests issued by DHCP clients, returning the requested network configuration settings.

The DHCP client runs on a network device. It issues a request to a DHCP server to obtain an IP address for the network device, and other network settings.

The DHCP relay agent acts an intermediary, forwarding DHCP requests/response between DHCP clients/servers on different networks. This enables DHCP clients to use the services of DHCP servers that are not on the same subnet on which they are located.

DHCP client

By default, the switch operates as a DHCP client on the management interface allowing it to automatically obtain an IP address from a DHCP server on the network to which it is connected.

DHCP client commands

ip dhcp

Syntax

```
ip dhcp
```

Description

Enables the DHCP client on the management interface enabling the interface to automatically obtain an IP address from a DHCP server on the network. By default, the DHCP client is enabled.

Command context

```
config-if-mgmt
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

This example enables the DHCP client on the management interface.

```
switch(config)# interface mgmt
switch(config-if-mgmt)# ip dhcp
switch(config-if-mgmt)# no shutdown
```

If the interface is not enabled, you can enable it by entering the `no shutdown` command.

DHCP relay agent

The function of the DHCP relay agent is to forward the DHCP messages to other subnets so that the DHCP server does not have to be on the same subnet as the DHCP clients. The DHCP relay agent transfers DHCP messages from the DHCP clients located on a subnet without a DHCP server, to other subnets. It also relays answers from DHCP servers to DHCP clients.

Supported interfaces

The DHCP relay agent is supported on layer 3 interfaces, layer 3 VLAN interfaces, and LAG interfaces. DHCP relay is not supported on the management interface.

VRF support

The DHCP relay agent is VRF aware and behaves as follows when VRFs are defined on the switch:

- DHCP client requests received on an interface are forwarded to the configured servers via the VRF that the interface is part of.
- DHCP server responses received on an interface are forwarded to the client that is reachable via the VRF that the interface is part of.

DHCPv4 relay agent

Hop count in DHCP requests

When a DHCP client broadcasts request, the DHCP relay agent in the switch receives the packets and forwards them to the DHCP server as unicast requests. During this process, the DHCP relay agent increments the hop count before forwarding DHCP packets to the server. The DHCP server, in turn, includes the hop count in the DHCP header in the response sent back to a DHCP client.

DHCP relay option 82

Option 82 is called the relay agent information option. When a DHCP relay agent forwards client-originated DHCP packets to a DHCP server, the option 82 field is inserted/replaced, or the packet with this option is dropped. Servers recognizing the relay agent information option may use the information to implement policies for the assignment of IP addresses and other parameters. The relay agent relays the server-to-client replies to the client.

If a second relay agent is configured to add its own option 82 information, it can encapsulate option 82 information in messages from a first relay agent. The DHCP server uses the option 82 information from both relay agents to decide the IP address for the client..

Inter-VRF DHCP relay

The DHCP relay agent supports anycast gateway using option 82 sub-option 5 (RFC 3527). The DHCP relay discovery packet is filled with the client's gateway IP address in sub-option 5 (discovery packet). The DHCP server uses this information to offer an IP address from the right pool. Pool selection occurs by matching the default gateway configuration settings on the DHCP server with the requested gateway IP address in sub-option 5 in the discovery packet.

The switch uses DHCP relay sub-option 151 to enable DHCP relay to forward discovery and reply packets between VXLAN DHCP clients and DHCP servers even when they are on different overlay or underlay VRFs and the DHCP-server is reachable on the default VRF or one of the overlay VRFs.

In general deployments, a renewal of a DHCP client's IP occurs when the client sends a request to the DHCP server directly. In the case of EVPN VXLAN clients, the DHCP server is not directly reachable. Instead, the renewal request is sent to the DHCP relay. DHCP relay agent fills the option 82 sub-option 11 field in the DHCP discovery packet with the client's gateway IP on the VTEP (which is the relay interface IP address of the VTEP) and the DHCP server returns a DHCP offer reply packet with option 54 set to the DHCP server

Identifier. When the reply packet is received by the client, the client uses the IP in option 54 to sent subsequent renewal requests to this IP (VTEP's Relay Interface IP) using sub-option 11 (also known as the Server ID Override Sub-option). Refer to RFC 5107 for more details.

Sub-options 5,11,151,152 are filled in the discover packet, only if a source IP address is defined (using the command `ip source-address`) for the given DHCP server's source VRF. If the server does not understand sub-option 151, then the server will add sub-option 152 in offer packet.

Configuring a BOOTP/DHCP relay gateway

The DHCP relay agent selects the lowest-numbered IP address on the interface to use for DHCP messages. The DHCP server then uses this IP address when it assigns client addresses. However, this IP address may not be the same subnet as the one on which the client needs the DHCP service. This feature provides a way to configure a gateway address for the DHCP relay agent to use for relayed DHCP requests, rather than the DHCP relay agent automatically assigning the lowest-numbered IP address.

Configuring the DHCPv4 relay agent

Prerequisites

An enabled layer 3 interface.

Procedure

1. The DHCPv4 relay agent is enabled by default. If it was previously disabled, enable it with the command **`dhcp-relay`**.
2. Configure one or more IP helper addresses with the command **`ip helper-address`**. This determines where the DHCPv4 agent forwards DHCP requests. IP helper addresses can be configured on layer 3 interfaces, layer 3 VLAN interfaces, and LAG interfaces.
3. If you want to modify the content of forwarded DHCP packets or drop DHCP packets, configure option 82 support with the command **`dhcp-relay option 82`**.
4. Define the gateway address that the DHCPv4 agent will use with the command **`ip bootp-gateway`**.
5. If required, enable the hop count increment feature with the command **`dhcp-relay hop-count-increment`**.
6. Review DHCPv4 relay agent configuration settings with the commands **`show dhcp-relay`**, **`show ip helper-address`**, and **`show dhcp-relay bootp-gateway`**.

Example

This example creates the following configuration:

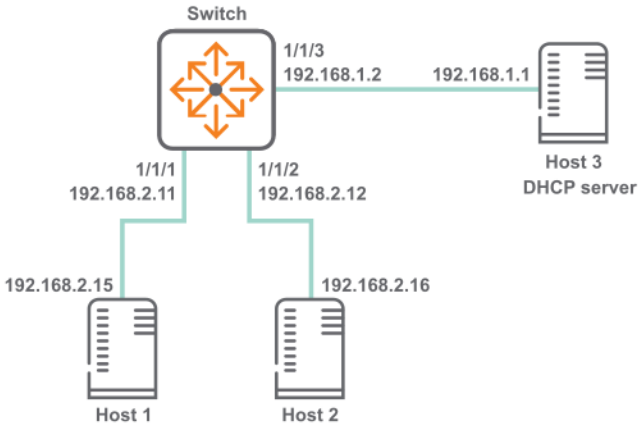
- Enables the DHCPv4 relay agent.
- Enables interface **`1/1/1`** and assigns an IPv4 address to it. (By default, all interfaces are layer 3 and disabled.)
- Defines an IP helper address of **`10.10.20.209`** on the interface.
- Enables DHCP option 82 support and replaces all option 82 information with the values from the switch with the switch MAC address as the remote ID.

```
switch(config)# dhcp-relay
switch(config)# interface 1/1/1
switch(config-if)# no shutdown
```

```
switch(config-if)# ip address 198.51.100.1/24
switch(config-if)# ip helper-address 10.10.20.209
switch(config-if)# exit
switch(config)# dhcp-relay option 82 replace mac
switch# show dhcp-relay
 DHCP Relay Agent                  : Enabled
 DHCP Request Hop Count Increment  : Enabled
 Option 82                         : Disabled
 Response Validation               : Disabled
 Option 82 Handle Policy           : replace
 Remote ID                         : mac

 DHCP Relay Statistics:

  Valid Requests  Dropped Requests  Valid Responses  Dropped Responses
  --------------  ----------------  ---------------  -----------------
  60              10                60               10

 DHCP Relay Option 82 Statistics:

  Valid Requests  Dropped Requests  Valid Responses  Dropped Responses
  --------------  ----------------  ---------------  -----------------
  50              8                 50               8
```

DHCPv4 relay scenario 1

In this scenario, DHCP relay on the server enables two hosts to obtain their IP addresses from a DHCP server on a different subnet. The physical topology of the network looks like this:



Procedure

1. DHCP relay is enabled by default. If it was previously disabled, enable it.

   ```
   switch# config
   switch(config)# dhcp-relay
   ```

2. Define an IPv4 helper address on interfaces `1/1/1` and `1/1/2` .

   ```
   switch(config)# interface 1/1/1
   switch(config-if)# ip address 192.168.2.11/24
   switch(config-if)# ip helper-address 192.168.1.1
   switch(config-if)# interface 1/1/2
   switch(config-if)# ip address 192.168.2.12/24
   switch(config-if)# ip helper-address 192.168.1.1
   switch(config-if)# quit
   ```

3. Verify DHCP relay configuration.

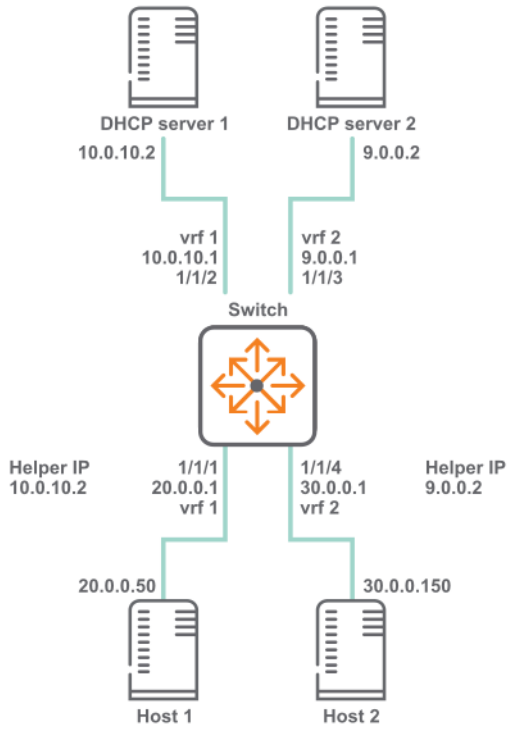
```
switch# show dhcp-relay

DHCP Relay Agent                  : Enabled
 DHCP Request Hop Count Increment : Enabled
 L2VPN Clients                    : Disabled
 Option 82                        : Disabled
 Source-Interface                 : Disabled
 Response Validation              : Disabled
 Option 82 Handle Policy          : replace
 Remote ID                        : mac

 DHCP Relay Statistics:

  Valid Requests Dropped Requests Valid Responses Dropped Responses
  -------------- ---------------- --------------- -----------------
  60             10               60              10

 DHCP Relay Option 82 Statistics:

  Valid Requests Dropped Requests Valid Responses Dropped Responses
  -------------- ---------------- --------------- -----------------
  50             8                50              8
```

```
switch# show ip helper-address
 IP Helper Addresses

 Interface: 1/1/1
  IP Helper Address     VRF
  -----------------     -----------------
  192.168.1.1           default

 Interface: 1/1/2
  IP Helper Address     VRF
  -----------------     -----------------
  192.168.1.1           default
```

DHCPv4 relay scenario 2

(This scenario is not supported on the 6200 Switch Series.)

In this scenario, the two host computers communicate with two different DHCP servers. Each server is reached on a different VRF. The physical topology of the network looks like this:



Procedure

1. Create the two VRFs.

```
switch# config
switch(config)# vrf vrf 1
switch(config)# vrf vrf 2
```

2. Configure interface `1/1/1`. Set its IP address, associate it with VRF 1, and define the helper IP address to reach DHCP server 1.

```
switch(configif)# interface 1/1/1
switch(configif)# vrf attach vrf1
switch(configif)# ip address 20.0.0.1/8
switch(configif)# ip helper-address 10.0.10.2
```

3. Configure interface `1/1/2`. Set its IP address and associate it with VRF 1.

```
switch(configif)# interface 1/1/2
switch(configif)# vrf attach vrf1
switch(configif)# ip address 10.0.10.1/24
```

4. Configure interface `1/1/3`. Set its IP address and associate it with VRF 1.

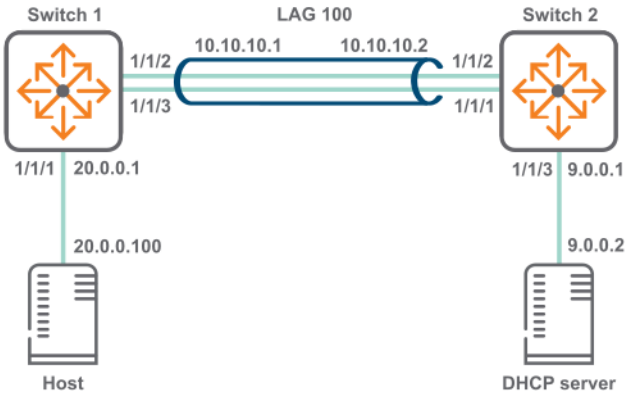
```
switch(configif)# interface 1/1/3
switch(configif)# vrf attach vrf2
switch(configif)# ip address 9.0.0.1/24
```

5. Configure interface `1/1/4`. Set its IP address, associate it with VRF 2, and define the helper IP address to reach DHCP server 2.

```
switch(configif)# interface 1/1/4
switch(configif)# vrf attach vrf2
switch(configif)# ip address 30.0.0.1/8
switch(configif)# ip helper-address 9.0.0.2
```

DHCPv4 relay scenario 3

In this scenario, host on switch 1 reaches the DHCP server on switch two via a LAG. The physical topology of the network looks like this:



Procedure

1. On switch 1:

   a. Create LAG **100** and assign an IP address to it.

   ```
   switch# config
   switch(config)# interface lag 100
   switch(config-lag-if)# ip address 10.0.10.1/24
   switch(config-lag-if)# lacp mode active
   switch(config-lag-if)# exit
   switch(config)#
   ```

   b. Assign an IP address to interface **1/1/1** and a an IP helper address to reach the DHCP server.

   ```
   switch(config)# interface 1/1/1
   switch(config-if)# ip address 20.0.0.1/8
   switch(config-if)# ip helper-address 9.0.0.2
   ```

   c. Assign interfaces **1/1/2** and **1/1/3** to LAG **100**

   ```
   switch(config-if)# interface 1/1/2
   switch(config-if)# lag 100
   switch(config-if)# interface 1/1/3
   switch(config-if)# lag 100
   switch(config-if)# exit
   switch(config)#
   ```

   d. Create a route between **10.0.10.2** and **9.0.0.0**.

   ```
   switch(config)# ip route 9.0.0.0/24 10.0.10.2
   ```

2. On switch 2:

   a. Create LAG **100** and assign an IP address to it.

   ```
   switch# config
   switch(config)# interface lag 100
   switch(config-lag-if)# ip address 10.0.10.2/24
   switch(config-lag-if)# lacp mode active
   ```

```
switch(config-lag-if)# exit
switch(config)#
```

b. Assign interfaces **1/1/1** and **1/1/2** to LAG **100**

```
switch(config-if)# interface 1/1/2
switch(config-if)# lag 100
switch(config-if)# interface 1/1/3
switch(config-if)# lag 100
switch(config-if)# exit
switch(config)#
```

c. Assign an IP address to interface **1/1/3**.

```
switch(config)# interface 1/1/3
switch(config-if)# ip address 9.0.0.1/24
```

d. Create a route between **20.0.0.0** and **10.0.10.1**.

```
switch(config)# ip route 20.0.0.0/8 10.0.10.1
```

DHCPv4 relay commands

dhcp-relay

Syntax

```
dhcp-relay

no dhcp-relay
```

Description

Enables DHCP relay support. DHCP relay is enabled by default. DHCP relay is not supported on the management interface.

The `no` form of this command disables DHCP relay support.

Command context

`config`

Authority

Administrators or local user group members with execution rights for this command.

Examples

This example enables DHCP relay support.

```
switch(config)# dhcp-relay
```

This example removes DHCP relay support.

```
switch(config)# no dhcp-relay
```

dhcp-relay hop-count-increment

Syntax

```
dhcp-relay hop-count-increment

no dhcp-relay hop-count-increment
```

Description

Enables the DHCP relay hop count increment feature, which causes the DHCP relay agent to increment the hop count in all relayed DHCP packets. Hop count is enabled by default.

The `no` form of this command disables the hop count increment feature.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling the hop count increment feature.

```
switch(config)# dhcp-relay hop-count-increment
```

Disabling the hop count increment feature.

```
switch(config)# no dhcp-relay hop-count-increment
```

dhcp-relay l2vpn-clients

Syntax

```
dhcp-relay l2vpn-clients

no dhcp-relay l2vpn-clients
```

Description

Enables forwarding of packets from L2 VPN clients. Forwarding is enabled by default.

The `no` form of this command disables forwarding of packets from L2 VPN clients.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Example

Enabling forwarding of packets from L2 VPN clients.

```
switch(config)# dhcp-relay l2vpn-clients
switch(config)# no dhcp-relay l2vpn-clients
```

dhcp-relay option 82

Syntax

```
dhcp-relay option 82 {replace [validate] | drop [validate] | keep | source-interface | validate [replace | drop]} [ip | mac]

no dhcp-relay option 82 source-interface
```

Description

Configures the behavior of DHCP relay option 82. A DHCP relay agent can receive a message from another DHCP relay agent having option 82. The relay information from the previous relay agent is replaced by default.

The `no` form of this command disables support for DHCP relay option 82.

Command context

`config`

Parameters

`replace`

Replace the existing option 82 field in an inbound client DHCP packet with the information from the switch. The remote ID and circuit ID information from the first relay agent is lost. Default.

`validate`

Validate option 82 information in DHCP server responses and drop invalid responses.

`drop`

Drop any inbound client DHCP packet that contains option 82 information.

`keep`

Keep the existing option 82 field in an inbound client DHCP packet. The remote ID and circuit ID information from the first relay agent is preserved.

`source-interface`

Configures the DHCP relay to use a configured source IP address for inter-VRF server reachability. Set the source IP address with the command `ip source-interface`.

`ip`

Use the IP address of the interface on which the client DHCP packet entered the switch as the option 82 remote ID.

`mac`

Use the MAC address of the switch as the option 82 remote ID. Default.

Authority

Administrators or local user group members with execution rights for this command.

Example

This example enables DHCP option 82 support and replaces all option 82 information with the values from the switch, with the switch MAC address as the remote ID.

```
switch(config)# dhcp-relay option 82 replace mac
```

ip bootp-gateway

Syntax

```
ip bootp-gateway <IPV4-ADDR>

no ip bootp-gateway <IPV4-ADDR>
```

Description

Configures a gateway address for the DHCP relay agent to use for DHCP requests. By default DHCP relay agent picks the lowest-numbered IP address on the interface.

The `no` form of this command removes the gateway address.

Command context

```
config-if
```

Parameters

<IPV4-ADDR>

Specifies the IP address of the gateway in IPv4 format (`x.x.x.x`), where `x` is a is a decimal number from 0 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Sets the IP address of the gateway for interface **`1/1/1`** to **`10.10.10.10`**.

```
switch(config)# interface 1/1/1
switch(config-if)# ip bootp-gateway 10.10.10.10
```

ip helper-address

Syntax

```
ip helper-address <IPV4-ADDR> [vrf <VRF-NAME>]

no ip helper-address <IPV4-ADDR> [vrf <VRF-NAME>]
```

Description

Defines the address of a remote DHCP server or DHCP relay agent. Up to eight addresses can be defined. The DHCP agent forwards DHCP client requests to all defined servers.

This command requires that you define a source IP address for DHCP relay with the command `ip source-interface`. The configured source IP on the VRF is used to forward DHCP packets to the server.

A helper address cannot be defined on the OOBM interface.

The `no` form of this command removes an IP helper address.

Command context

```
config-if
```

Parameters

`helper-address <IPV4-ADDR>`

Specifies the helper IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

`vrf <VRF-NAME>`

Specifies the name of a VRF. Default: `default`.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defining the IP helper address **`10.10.10.209`** on interface **`1/1/1`**.

```
switch(config)# interface 1/1/1
switch(config-if)# ip helper-address 10.10.10.209
```

Removing the IP helper address **`10.10.10.209`** on interface **`1/1/1`**.

```
switch(config-if)# no ip helper-address 10.10.10.209
```

Defining the IP helper address **`10.10.10.209`** on interface **`1/1/2`** on VRF `myvrf` .

```
switch(config)# interface 1/1/2
switch(config-if)# ip helper-address 10.10.10.209 vrf myvrf
```

Removing the IP helper address **`10.10.10.209`** on interface **`1/1/2`** on VRF `myvrf`.

```
switch(config-if)# no ip helper-address 10.10.10.209 vrf myvrf
```

show dhcp-relay

Syntax

```
show dhcp-relay [vsx-peer]
```

Description

Shows DHCP relay configuration settings.

Command context

Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Parameters

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Example

```
switch# show dhcp-relay

DHCP Relay Agent                  : Enabled
 DHCP Request Hop Count Increment : Enabled
 L2VPN Clients                    : Disabled
 Option 82                        : Disabled
 Source-Interface                 : Disabled
 Response Validation              : Disabled
 Option 82 Handle Policy          : replace
 Remote ID                        : mac

 DHCP Relay Statistics:

  Valid Requests Dropped Requests Valid Responses Dropped Responses
  -------------- ---------------- --------------- -----------------
  60             10               60              10

 DHCP Relay Option 82 Statistics:

  Valid Requests Dropped Requests Valid Responses Dropped Responses
  -------------- ---------------- --------------- -----------------
  50             8                50              8
```

show dhcp-relay bootp-gateway

Syntax

```
show dhcp-relay bootp-gateway [interface <INTERFACE-NAME>] [vsx-peer]
```

Description

Shows the bootp gateway defined for all interfaces or a specific interface.

Command context

Manager (#)

Parameters

<INTERFACE-NAME>

Specifies an interface. Format: `member/slot/port`.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show dhcp-relay bootp-gateway

 BOOTP Gateway Entries

 Interface           Source IP
 ------------------- ---------------
```

```
 1/1/1                1.1.1.1
 1/1/2                1.1.1.2
```

```
switch# show ip helper-address interface 1/1/1

 BOOTP Gateway Entries

 Interface            Source IP
 -------------------- ---------------
 1/1/1                1.1.1.1
```

show ip helper-address

Syntax

```
show ip helper-address [interface <INTERFACE-ID>] [vsx-peer]
```

Description

Shows the helper IP addresses defined for all interfaces or a specific interface.

Command context

Manager (#)

Parameters

interface *<INTERFACE-ID>*

Specifies an interface. Format: `member/slot/port`.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show ip helper-address
 IP Helper Addresses

 Interface: 1/1/1
  IP Helper Address     VRF
  ----------------      ----------------
  192.168.20.1          default
  192.168.10.1          default

 Interface: 1/1/2
  IP Helper Address     VRF
  ----------------      ----------------
  192.168.30.1          RED
```

```
switch# show ip helper-address interface 1/1/1
 IP Helper Addresses

 Interface: 1/1/1
  IP Helper Address     VRF
```

```
-----------------     -----------------
192.168.20.1          default
192.168.10.1          default
```

DHCPv6 relay agent

Supporting VXLAN topologies or inter-VRF deployment

When deploying EVPN VXLAN or inter-VRF topologies where the source VRFs for the DHCP and DHCP client are different, it is recommended that you install the DHCPv6 server in the underlay so that there is only one instance of the DHCPv6 server serving overlay clients.

Configuring the DHCPv6 relay agent

Prerequisites

An enabled layer 3 interface.

Procedure

1. Enable the DHCPv6 agent with the command **dhcpv6-relay**.
2. Configure one or more IP helper addresses with the command **ipv6 helper-address**. This determines where the DHCPv6 agent forward DHCP requests.
3. If you want to enable DHCP option 79 support to forward client link-layer addresses, use the command **dhcpv6-relay option 79**.
4. Review DHCPv6 relay agent configuration settings with the commands **show dhcpv6-relay** and **show ipv6 helper-address**.

Example

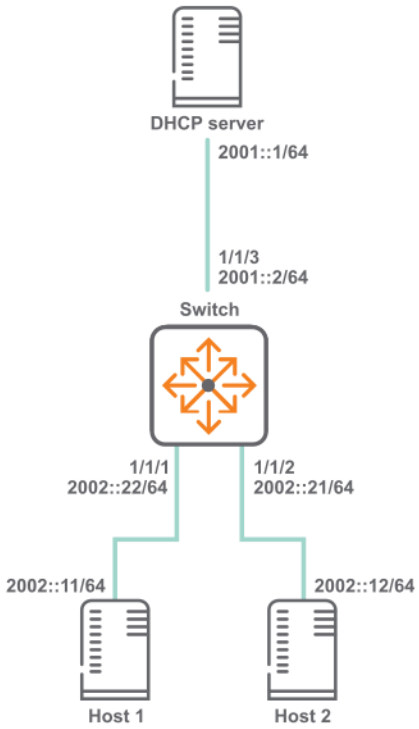
This example creates the following configuration:

- Enables the DHCPv6 relay agent.
- Enables interface **1/1/2** and assigns an IPv6 address to it. (By default, all interfaces are layer 3 and disabled.)
- Defines an IP helper address of **FF01::1:1000** on interface **1/1/2**.
- Enables DHCP option 79.

```
switch(config)# dhcpv6-relay
switch(config)# interface 1/1/2
switch(config-if)# no shutdown
switch(config-if)# ipv6 address 2001:0db8:85a3::8a2e:0370:7334/24
switch(config-if)# ip helper-address FF01::1:1000
switch(config-if)# exit
switch(config)# dhcpv6-relay option 79
```

DHCPv6 relay scenario 1

In this scenario, DHCP relay on the server enables two hosts to obtain their IP addresses from a DHCP server on a different subnet. The physical topology of the network looks like this:



Procedure

1. Enable DHCP relay.

```
switch# config
switch(config)# dhcpv6-relay
```

2. Define an IPv6 helper address on interfaces `1/1/1` and `1/1/2` .

```
switch(config)# interface 1/1/1
switch(config-if)# ipv6 address 2002::22/64
switch(config-if)# ipv6 helper-address 2001::1
switch(config-if)# interface 1/1/2
switch(config-if)# ipv6 address 2002::21/64
switch(config-if)# ipv6 helper-address 2001::1
switch(config-if)# quit
```

3. Verify DHCP relay configuration.

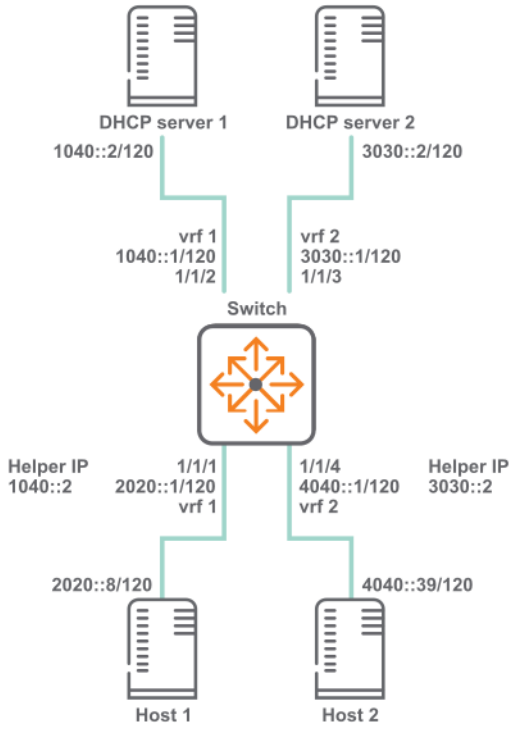
```
switch# show dhcpv6-relay
  DHCPv6 Relay Agent : Enabled
  Option 79          : Disabled
```

```
switch# show ipv6 helper-address

 Interface: 1/1/1
 IPv6 Helper Address                            Egress Port
 ---------------------------------------------- -----------
 2001::1                                        1/1/3

 Interface: 1/1/2
 IPv6 Helper Address                            Egress Port
 ---------------------------------------------  -----------
 2001::1                                        1/1/3
```

DHCPv6 relay scenario 2

(This scenario is not supported on the 6200 Switch Series.)

In this scenario, the two host computers communicate with two different DHCP servers. Each server is reached on a different VRF. The physical topology of the network looks like this:



Procedure

1. Create the two VRFs.

```
switch# config
switch(config)# vrf vrf 1
switch(config)# vrf vrf 2
```

2. Configure interface `1/1/1`. Set its IP address, associate it with VRF 1, and define the helper IP address to reach DHCP server 1.

```
switch(configif)# interface 1/1/1
switch(configif)# vrf attach vrf1
switch(configif)# ipv6 address 20.0.0.1/8
switch(configif)# ipv6 helper-address unicast 1040::2
```

3. Configure interface `1/1/2`. Set its IP address and associate it with VRF 1.

```
switch(configif)# interface 1/1/2
switch(configif)# vrf attach vrf1
switch(configif)# ipv6 address  1040::1/120
```

4. Configure interface **1/1/3**. Set its IP address and associate it with VRF 1.

```
switch(configif)# interface 1/1/3
switch(configif)# vrf attach vrf2
switch(configif)# ipv6 address 3030::1/120
```

5. Configure interface **1/1/4**. Set its IP address, associate it with VRF 2, and define the helper IP address to reach DHCP server 2.

```
switch(configif)# interface 1/1/4
switch(configif)# vrf attach vrf2
switch(configif)# ipv6 address  4040::1/120
switch(configif)# ipv6 helper-address unicast 3030::2
```

DHCP relay (IPv6) commands

dhcpv6-relay

Syntax

```
dhcpv6-relay

no dhcpv6-relay
```

Description

Enables DHCPv6 relay support. DHCPv6 relay is disabled by default.

DHCP relay is not supported on the management interface

The `no` form of this command disables DHCP relay support.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enables DHCPv6 relay support.

```
switch(config)# dhcpv6-relay
```

Removes DHCPv6 relay support.

```
switch(config)# no dhcpv6-relay
```

dhcpv6-relay option 79

Syntax

```
dhcpv6-relay option 79

no dhcpv6-relay option 79
```

Description

Enables support for DHCP relay option 79. When enabled, the DHCPv6 relay agent forwards the link-layer address of the client. This option is disabled by default.

The `no` form of this command disables support for DHCP relay option 79.

Command context

`config`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enables DHCP option 79 support.

```
switch(config)# dhcpv6-relay option 79
```

Disables DHCP option 79 support.

```
switch(config)# no dhcpv6-relay option 79
```

ipv6 helper-address

Syntax

`ipv6 helper-address unicast <UNICAST-IPV6-ADDR>`

`no ipv6 helper-address unicast <UNICAST-IPV6-ADDR>`

`ipv6 helper-address multicast {all-dhcp-servers | <MULTICAST-IPV6-ADDR>} egress <PORT-NUM>`

`no ipv6 helper-address multicast {all-dhcp-servers | <MULTICAST-IPV6-ADDR>} egress <PORT-NUM>`

Description

Defines the address of a remote DHCPv6 server or DHCPv6 relay agent. Up to eight addresses can be defined. The DHCPv6 agent forwards DHCPv6 client requests to all defined servers.

Not supported on the OOBM interface.

The `no` form of this command removes an IP helper address.

Command context

`config-if`

Parameters

<UNICAST-IPV6-ADDR>

Specifies the unicast helper IP address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

<MULTICAST-IPV6-ADDR>

Specifies the multicast helper IP address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

all-dhcp-servers

Specifies all the DHCP server IPv6 addresses for the interface.

egress *<PORT-NUM>*

Specifies the port number on which DHCPv6 service requests are relayed to a multicast destination. The egress port must be different than the one on which the multicast helper address is configured. Format: `member/slot/port`.

vrf *<VRF-NAME>*

Specifies the name of the VRF from which the specified protocol sets its source IP address.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defining a multicast IPv6 helper address of **2001:DB8::1** on port **1/1/2**:

```
switch(config-if)# ipv6 helper-address multicast 2001:DB8:0:0:0:0:0:1 egress 1/1/2
```

Removing the IP helper address of **2001:DB8::1** on port **1/1/2**:

```
switch(config-if)# no ipv6 helper-address multicast 2001:DB8:0:0:0:0:0:1 egress 1/1/2
```

show dhcpv6-relay

Syntax

```
show dhcpv6-relay [vsx-peer]
```

Description

Shows DHCP relay configuration settings.

Command context

Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show dhcpv6-relay
  DHCPv6 Relay Agent : Enabled
  Option 79          : Enabled
```

show ipv6 helper-address

Syntax

```
show ipv6 helper-address [interface <INTERFACE-ID>] [vsx-peer]
```

Description

Shows the helper IP addresses defined for all interfaces or a specific interface.

Command context

Manager (#)

Parameters

`interface <INTERFACE-ID>`

Specifies an interface. Format: `member/slot/port`.

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show ipv6 helper-address

 Interface: 1/1/1
 IPv6 Helper Address                          Egress Port
 -------------------------------------------- -----------
 2001:db8:0:1::                               -
 FF01::1:1000                                 1/1/2

 Interface: 1/1/2
 IPv6 Helper Address                          Egress Port
 -------------------------------------------  -----------
 2001:db8:0:1::                               -

switch# show ipv6 helper-address interface 1/1/1

 Interface: 1/1/1
 IPv6 Helper Address                          Egress Port
 -------------------------------------------- -----------
 2001:db8:0:1::                               -
 FF01::1:1000                                 1/1/2
```

```
switch# show ipv6 helper-address interface 1/1/1

 Interface: 1/1/1
 IPv6 Helper Address                          Egress Port
 -------------------------------------------- -----------
 2001:db8:0:1::                               -
 FF01::1:1000                                 1/1/2
```

DHCP server

Overview

The dynamic host configuration protocol (DHCP) enables a server to automate the assignment of IP addresses, and other networking settings, to host computers. The DHCP server on the switch provides both IPv4 and IPv6 support and is independently configurable on each VRF.

Key features

- Supports multiple address pools and static address bindings.
- Supports DHCP options, enabling the server to provide additional information about the network when DHCP clients request an address.
- Supports BOOTP to distribute boot image files using an external TFTP server.
- VRF aware, meaning that DHCP client requests received on an interface are processed by the DHCP server instance configured for a VRF. DHCP server responses are forwarded to clients on the VRF.
- Supports external storage of lease information on a remote host. This enables the DHCP server to restore lease information after a reboot or a failure. Lease information is stored in a flat file on the configured external device. It is important that the external device provide persistent external storage to allow restoration of lease information. If external storage is not configured, then after a failure or reboot, all existing lease information is lost.
- Supports VSX. In a VSX setup, one switch acts as primary and the other switch acts as secondary. The DHCP server is active only on the primary switch. After a failover, the DHCP server is enabled based on the state and role of the switch. The state of the DHCP server indicates the operational state of the server. VSX synchronization supports DHCPv4 and DHCPv6 server, including external storage configurations. For more information on VSX support, see the *ArubaOS-CX Virtual Switching Extension (VSX) Guide*.

DHCP relay interoperation

DHCP server and DHCP relay cannot both be active on interfaces belonging to the same VRF.

Configuring a DHCPv4 server on a VRF

Prerequisites

- An enabled layer 3 interface.
- A VRF.
- An external TFTP server to host BOOTP image files (optional).
- An external storage device installed and configured (optional).

Procedure

1. Assign the DHCPv4 server to a VRF with the command **`dhcp-server vrf`**. This switches to the DHCPv4 server configuration context.
2. If you want the DHCPv4 server to be the sole authority for IP addresses on the VRF, enable authoritative mode with the command **`authoritative`**.
3. Define an address pool for the VRF with the command **`pool`**. This switches to the DHCPv4 server pool context. Customize pool settings as follows:

a. Define the range of addresses in the pool with the command **range**.

b. Set the lease time for addresses in the pool with the command **lease**.

c. Set the domain name for the pool with the command **domain-name**.

d. Define up to four default routers with the command **default-router**.

e. Define up to four DNS servers with the command **dns-server**.

f. Create static bindings for specific addresses in the pool with the command **static-bind**.

g. Configure custom DHCPv4 options for the pool with the command **option**.

h. Configure NetBIOS support with the commands **netbios-name-server** and **netbios-node-type**.

i. Configure BOOTP options with the command **bootp**.

j. Exit the DHCPv4 server pool context with the command **exit**.

4. Enable the DHCP server on the VRF with the command **enable**.
5. Configure support for persistent external storage of DHCP settings with the command **dhcp-server external-storage**.
6. View DHCPv4 server configuration settings with the command **show dhcp-server all-vrfs**.

Example

This example creates the following configuration:

- Configures the DHCPv4 server on VRF **primary-vrf**.
- Enables authoritative mode.
- Defines the pool **primary-pool** with the following settings:
  - Address range: **10.0.0.1** to **10.0.0.100**.
  - Lease time: 12 hours.
  - Domain name: **example.org.in**.
  - Default routers: **10.30.30.1** and **10.30.30.2**.
  - DNS servers: **125.0.0.1** and **125.0.0.2**.
  - Static binding of **10.0.0.11** for MAC address **24:be:05:24:75:73**.
  - DHCP custom option **3** with IP address **10.30.30.3**.
- Enables the DHCPv4 server.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# range 10.0.0.1 10.0.0.100
switch(config-dhcp-server-pool)# lease 12:00:00
switch(config-dhcp-server-pool)# domain-name example.org.in
switch(config-dhcp-server-pool)# default-router ip 10.30.30.1 10.30.30.2
switch(config-dhcp-server-pool)# dns-server 125.0.0.1 125.0.0.2
switch(config-dhcp-server-pool)# static-bind ip 10.0.0.11 mac 24:be:05:24:75:73
switch(config-dhcp-server-pool)# option 3 ip 10.30.30.3
```

```
switch(config-dhcp-server-pool)# exit
switch(config-dhcp-server)# enable
```

Configuring the DHCPv6 server on a VRF

Prerequisites

- An enabled layer 3 interface.
- A VRF.
- An external storage device installed and configured (optional).

Procedure

1. Assign the DHCPv6 server to a VRF with the command **dhcpv6-server vrf**. This switches to the DHCPv6 server configuration context.
2. If you want the DHCP server to be the sole authority for IP addresses on the VRF, enable authoritative mode with the command **authoritative**.
3. Define an address pool for the VRF with the command **pool**. This switches to the DHCPv6 server pool context. Customize pool settings as follows:
   a. Define the range of addresses in the pool with the command **range**.
   b. Set the DHCP lease time for addresses in the pool with the command **lease**.
   c. Define up to four DNS servers with the command **dns-server**.
   d. Create static bindings for specific addresses in the pool with the command **static-bind**.
   e. Configure custom DHCP options for the pool with the command **option**.
   f. Exit the DHCP server pool context with the command **exit**.
4. Enable the DHCPv6 server on the VRF with the command **enable**.
5. Configure support for persistent external storage of DHCP settings with the command **dhcv6p-server external-storage**.
6. View DHCPv6 server configuration settings with the command **show dhcpv6-server all-vrfs**.

Example

This example creates the following configuration:

- Configures a DHCPv6 server on VRF **primary-vrf**.
- Enables authoritative mode.
- Defines the pool **primary-pool** with the following settings:
  - Address range: **2001::1** to **2001::100**.
  - Lease time: 12 hours.
  - DNS servers: **2101::14** and **2101::14**.

- Static binding of **2001::101** for client ID **1:0:a0:24:ab:fb:9c**.
- DHCP custom option: **22** with IP address **2101::15**.

- Enables the DHCPv6 server.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# range 2001::1 2001::100 prefix-len 64
switch(config-dhcpv6-server-pool)# lease 12:00:00
switch(config-dhcpv6-server-pool)# dns-server 2101::13 2101::14
switch(config-dhcpv6-server-pool)# static-bind ipv6 2001::10 client-id 1:0:a0:24:ab:fb:9c
switch(config-dhcpv6-server-pool)# option 22 ipv6 2101::15
switch(config-dhcpv6-server-pool)# exit
switch(config-dhcpv6-server)# enable
```

DHCP server IPv4 commands

authoritative

Syntax

```
authoritative

no authoritative
```

Description

Configures the DHCPv4 server as *authoritative* on the current VRF. This means that the server is the sole authority for the network on the VRF. Therefore, if a client requests an IP address lease for which the server has no record, the server responds with DHCPNAK, indicating that the client must no longer use that IP address. If the server is not authoritative, then it will ignore DHCPv4 requests received for unknown leases from unknown hosts.

The `no` form of this command disables authoritative mode on the current VRF.

Command context

```
config-dhcp-server
```

Authority

Administrators or local user group members with execution rights for this command.

Example

Configures DHCPv4 server authoritative mode on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# authoritative
```

Removes the DHCPv4 server authoritative mode on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# no authoritative
```

bootp

Syntax

```
bootp <REMOTE-URL>

no bootp <REMOTE-URL>
```

Description

Sets the BOOTP options that are returned by the DHCPv4 server for the current pool. BOOTP provides a way to distribute an IP address and boot image file to client stations. The DHCPv4 server returns the IP address and the location of the boot image file, which must be stored on an external TFTP server.

The `no` form of this command disables support for BOOTP.

Command context

`config-dhcp-server-pool`

Parameters

<REMOTE-URL>

Specifies the name and location of a BOOTP file on a TFTP server in the format:

`tftp://{<IP> | <HOST>}/<FILE>`

- *<IP>*: Specifies the IP address of the TFTP server hosting the file in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255. You can remove leading zeros. For example, the address `192.169.005.100` becomes `192.168.5.100`.
- *<HOST>*: Specifies the fully-qualified domain name of the TFTP server hosting the file. Range: 1 to 64 printable ASCII characters.
- *<FILE>*: Specifies the name of the BOOTP file. Range: 1 to 64 printable ASCII characters.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines BOOTP support on the DHCPv4 server pool **`primary-pool`** on VRF **`primary`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# bootp tftp://10.0.0.1/mybootfile
```

Deletes BOOTP support on the DHCPv4 server pool **`primary-pool`** on VRF **`primary`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no bootp tftp://10.0.0.1/mybootfile
```

clear dhcp-server leases

Syntax

```
clear dhcp-server leases [all-vrfs | <IPV4-ADDR> vrf <VRF-NAME>] | vrf <VRF-NAME>]
```

Description

Clears DHCPv4 server lease information. The DHCPv4 server must be disabled before clearing lease information.

Command context

Manager (#)

Parameters

`all-vrfs`

Clears leases for all VRFs.

`<IPV4-ADDR>` **`vrf`** ***`<VRF-NAME>`***

Clears the lease for a specific client on a specific VRF. Specify the client address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255. You can remove leading zeros. For example, the address `192.169.005.100` becomes `192.168.5.100`.

`vrf` ***`<VRF-NAME>`***

Clears leases for a specific VRF.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Clearing all DHCPv4 server leases.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# disable
switch(config-dhcp-server)# exit
switch(config)# exit
switch# clear dhcp-server leases
```

Clearing all DHCPv4 server leases for VRF **`primary-vrf`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# disable
switch(config-dhcp-server)# exit
switch(config)# exit
switch# clear dhcp-server leases vrf primary-vrf
```

Clear the DHCPv4 server lease for IP address **`10.10.10.1`** on VRF **`primary-vrf`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# disable
switch(config-dhcp-server)# exit
switch(config)# exit
switch# clear dhcp-server leases 10.10.10.1 vrf primary-vrf
```

default-router

Syntax

```
default-router <IPV4-ADDR-LIST>

no default-router <IPV4-ADDR-LIST>
```

Description

Defines up to four default routers for the current DHCPv4 server pool.

The `no` form of this command removes the specified default routers from the pool.

Command context

`config-dhcp-server-pool`

Parameters

<IPV4-ADDR-LIST>

Specifies the IP addresses of the default routers in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255. You can remove leading zeros. For example, the address `192.169.005.100` becomes `192.168.5.100`. Separate addresses with a space. A maximum of four IP addresses can be defined.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines two default routers, **`10.0.0.1`** and **`10.0.0.10`**, for the server pool **`primary-pool`** on VRF **`primary`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# default-router ip 10.0.0.1 10.0.0.10
```

Deletes the default router **`10.0.0.1`** from the server pool **`primary-pool`** on VRF **`primary`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no default-router ip 10.0.0.1
```

dhcp-server external-storage

Syntax

```
dhcp-server external-storage <VOLUME-NAME> file <LEASE-FILENAME> [delay <DELAY>]

no dhcp-server external-storage <VOLUME-NAME> file <LEASE-FILENAME> [delay <DELAY>]
```

Description

Configures the external storage file location for DHCPv4 server lease information. This file provides persistent storage, enabling DHCPv4 server settings to be restored when the switch is restarted. Lease information is stored in a flat file on the configured external device.

If external storage is not configured, then after a failure or reboot, all existing lease information is lost.

Lease information is saved to external storage each time the delay timer expires, which by default is every 300 seconds.

Lease information is not restored when issuing the command `dhcp-server enable`.

The `no` form of this command removes external storage support for the DHCPv4 server.

Command context

`config`

Parameters

<VOLUME-NAME>

Specifies the external storage volume name. Range: 1 to 64 printable ASCII characters.

file *<LEASE-FILENAME>*

Specifies the external storage filename. Range: 1 to 255 printable ASCII characters.

delay *<DELAY>*

Specifies the interval in seconds between updates to the external storage file. Range: 15 to 86400. Default: 300.

Authority

Administrators or local user group members with execution rights for this command.

Example

Stores the lease file on external storage volume **Storage1** in file **LeaseFile** at an interval of 600 seconds.

```
switch(config)# dhcp-server external-storage Storage1 file LeaseFile delay 600
```

Disables storage of the lease file on external storage volume **Storage1** in file **LeaseFile**.

```
switch(config)# no dhcp-server external-storage Storage1 file LeaseFile delay 600
```

dhcp-server vrf

Syntax

```
dhcp-server vrf VRF-NAME

no dhcp-server vrf VRF-NAME
```

Description

Configures the DHCPv4 server to support a VRF and changes to the `config-dhcp-server` context for that VRF.

The `no` form of this command removes DHCPv4 server support on a VRF.

Command context

`config`

Parameters

VRF-NAME

Name of a VRF.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configures DHCPv4 server support on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
```

Removes DHCPv4 server support on VRF **primary**.

```
switch(config)# no dhcp-server vrf primary
```

disable

Syntax

```
disable
```

Description

Disables the DHCPv4 server on the current VRF. The DHCPv4 server is disabled by default when configured on a VRF.

Command context

```
config-dhcp-server
```

Authority

Administrators or local user group members with execution rights for this command.

Example

Disables the DHCPv4 server on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# disable
```

dns-server

Syntax

```
dns-server <IPV4-ADDR-LIST>

no dns-server <IPV4-ADDR-LIST>
```

Description

Defines up to four DNS servers for the current DHCPv4 server pool.

The `no` form of this command removes the specified DNS servers from the pool.

Command context

```
config-dhcp-server-pool
```

Parameters

<IPV4-ADDR-LIST>

Specifies the IP addresses of the DNS servers in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.Separate addresses with a space.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines two DNS servers for the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# dns-server 10.0.20.1
```

Deletes a DNS server from the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no dns-server 10.0.20.1
```

domain-name

Syntax

```
domain-name  <DOMAIN-NAME>

no domain-name  <DOMAIN-NAME>
```

Description

Defines a domain name for the current DHCPv4 server pool.

The `no` form of this command removes the specified domain name from the pool.

Command context

```
config-dhcp-server-pool
```

Parameters

<DOMAIN-NAME>

Specifies a domain name. Range: 1 to 255 printable ASCII characters.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines a domain name for the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# domain-name example.org.in
```

Deletes a domain name from the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no domain-name example.org.in
```

enable

Syntax

```
enable
```

Description

Enables the DHCPv4 server on the current VRF. The DHCPv4 server is disabled by default when configured on a VRF.

Command context

`config-dhcp-server`

Authority

Administrators or local user group members with execution rights for this command.

Example

Enables the DHCPv4 server on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# enable
```

lease

Syntax

```
lease {<TIME> | infinite}

no lease
```

Description

Sets the length of the DHCPv4 lease time for the current pool. The lease time determines how long an IP address is valid before a DHCPv4 client must request that it be renewed.

The `no` form of this command returns the DHCPv4 lease time to its default value 1 hour.

Command context

`config-dhcp-server-pool`

Parameters

<TIME>

Sets the DHCPv4 lease time. Format: DD:HH:MM. Default: 01:00:00.

infinite

Sets the DHCPv4 lease time to infinite. This means that addresses do not need to be renewed.

Authority

Administrators or local user group members with execution rights for this command.

Example

Sets the lease time for DHCPv4 server pool **primary-pool** on VRF **primary** to **12** hours.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# lease 00:12:00
```

Deletes the lease time for DHCPv4 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no lease 00:12:00
```

netbios-name-server

Syntax

```
netbios-name-server <IPV4-ADDR-LIST>

no netbios-name-server <IPV4-ADDR-LIST>
```

Description

Defines up to four NetBIOS WINS servers for the current DHCPv4 server pool. WINS is used by Microsoft DHCP clients to match host names with IP addresses.

The `no` form of this command removes the specified WINS servers from the pool.

Command context

```
config-dhcp-server-pool
```

Parameters

<IPV4-ADDR-LIST>

Specifies the IP addresses of NetBIOS (WINS) servers in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255. Separate addresses with a space. A maximum of four IP addresses can be defined.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines two WINS servers for the server pool **`primary-pool`** on VRF **`primary`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# netbios-name-server ip 10.0.20.1 10.0.30.10
```

Deletes a WINS server from the server pool **`primary-pool`** on VRF **`primary`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no netbios-name-server ip 10.0.20.1
```

netbios-node-type

Syntax

```
netbios-node-type <TYPE>

no netbios-node-type <TYPE>
```

Description

Defines the NetBIOS node type for the current DHCPv4 server pool.

The `no` form of this command removes the NetBIOS node type for the current pool.

Command context

```
config-dhcp-server-pool
```

Parameters

<TYPE>

Specifies the NetBIOS node type: `broadcast`, `hybrid`, `mixed`, or `peer-to-peer`.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines the NetBIOS node type **`broadcast`** for the DHCPv4 server pool **`primary-pool`** on VRF **`primary`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# netbios-node-type broadcast
```

Deletes the NetBIOS node type **`broadcast`** from the DHCPv4 server pool **`primary-pool`** on VRF **`primary`**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no netbios-node-type broadcast
```

option

Syntax

```
option <OPTION-NUM> {ascii <ASCII-STR> | hex <HEX-STR> | ip <IPV4-ADDR-LIST>}

no option <OPTION-NUM> {ascii <ASCII-STR> | hex <HEX-STR> | ip <IPV4-ADDR-LIST>}
```

Description

Defines custom DHCPv4 options for the current DHCPv4 server pool. DHCPv4 options enable the DHCPv4 server to provide additional information about the network when DHCPv4 clients request an address.

The `no` form of this command removes custom DHCPv4 options from the pool.

Command context

`config-dhcp-server-pool`

Parameters

<OPTION-NUM>

Specifies a DHCPv4 option number. For a list of DHCPv4 option numbers, see **https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml**. Range: 2 to 254.

`ascii` ***<ASCII-STR>***

Specifies a value for the selected option as an ASCII string. Range: 1 to 255 ASCII characters.

`hex` ***<HEX-STR>***

Specifies a value for the selected option as a hexadecimal string. Range: 1 to 255 hexadecimal characters.

`ip` ***<IPV4-ADDR-LIST>***

Specifies a list of IP addresses in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255. Separate addresses with a space. A maximum of four IP addresses can be defined.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines DHCPv4 option **3** for the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# option 3 ip 192.168.1.1
```

Deletes DHCPv4 option **3** for the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no option 3 ip 192.168.1.1
```

pool

Syntax

```
pool <POOL-NAME>

no pool <POOL-NAME>
```

Description

Creates a DHCPv4 server pool for the current VRF and switches to the `config-dhcp-server-pool` context for it. Multiple pools, each with a distinct range, can be assigned to a VRF. A maximum of 64 pools (IPv4 and IPv6), 64 address ranges, and 8182 clients are supported on the switch across all VRFs.

The `no` form of this command deletes the specified DHCPv4 server pool.

Command context

`config-dhcp-server`

Parameters

<POOL-NAME>

Specifies the DHCPv4 pool name. A maximum of 64 pools (IPv4 and IPv6) are supported across VRFs on the switch. Range: 1 to 32 printable ASCII characters. First character must be a letter or number.

Authority

Administrators or local user group members with execution rights for this command.

Example

Creates the DHCv4 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)#
```

Deletes the DHCPv4 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# no pool primary-pool
```

range

Syntax

```
range <LOW-IPV4-ADDR> <HIGH-IPV4-ADDR> [prefix-len <MASK>]

no range <LOW-IPV4-ADDR> <HIGH-IPV4-ADDR> [prefix-len <MASK>]
```

Description

Defines the range of IP addresses supported by the current DHCPv4 server pool. A maximum of 64 ranges are supported per switch across all VRFs.

The `no` form of this command deletes the address range for the current pool.

Command context

```
config-dhcp-server-pool
```

Parameters

<LOW-IPV4-ADDR>

Specifies the lowest IP address in the pool in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

<HIGH-IPV4-ADDR>

Specifies the highest IP address in the pool in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

prefix-len *<MASK>*

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 32.

NOTE: When active gateway is configured on the interface serviced by the pool, you must specify a prefix length that matches the mask on the IP address assigned to the interface. Otherwise, client stations will get a prefix length from active gateway that may not be consistent with the configured range, and a DHCP error will occur. In the following example, the DHCP range prefix is set to 16 to match the mask on the IP address assigned to interface VLAN 2.

```
switch(config)# interface vlan 2
switch(config-if-vlan)# ip address 200.1.1.1/16
switch(config-if-vlan)# active-gateway ip 200.1.1.3 mac
00:aa:aa:aa:aa:aa
switch(config-if-vlan)# exit
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# range 192.168.1.1 192.168.1.100
prefix-len 16
```

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines the address range **192.168.1.1** to **192.168.1.100** with a mask of **24** bits for the DHCPv4 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# 192.168.1.1 192.168.1.100 prefix-len 24
```

Deletes the address range **192.168.1.1** to **192.168.1.100** with a mask of **24** bits from the DHCPv4 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no 192.168.1.1 192.168.1.100 prefix-len 24
```

show dhcp-server

Syntax

```
show dhcp-server [all-vrfs]

show dhcp-server leases {all-vrfs | vrf <VRF-NAME>}

show dhcp-server pool <POOL-NAME> [vrf <VRF-NAME>]
```

Description

Shows configuration settings for the DHCPv4 server.

Command context

Manager (#)

Parameters

all-vrfs

Shows DHCPv4 server configuration settings for all VRFs.

leases {all-vrfs | vrf *<VRF-NAME>*}

Shows DHCPv4 server lease configuration settings for all VRFs or a specific VRF.

pool *<POOL-NAME>* [vrf *<VRF-NAME>*]

Shows DHCPv4 server pool configuration settings for all VRFs or a specific VRF.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing all DHCPv4 server configuration settings.

```
switch# show dhcp-server

VRF Name            : default
DHCP Server         : enabled
Operational State   : operational
Authoritative Mode  : false

Pool Name           : test
Lease Duration      : 00:01:00


DHCP dynamic IP allocation
--------------------------
Start-IP-Address    End-IP-Address    Prefix-Length
```

```
----------------   --------------   --------------
192.168.1.1        192.168.1.20     24


DHCP Server options
-------------------
Option-Number   Option-Type    Option-Value
-------------   -----------    ------------
6               ip             10.0.0.3 10.0.0.4 10.0.0.5 10.0.0.6


DHCP Server static IP allocation
--------------------------------
IP-Address     Client-Hostname      State            MAC-Address
-----------    ----------------     ------           ------------
10.0.0.3       *                    OPERATIONAL      aa:aa:aa:aa:aa:aa


BOOTP Options
---------------
Boot-File-Name    TFTP-Server-Name    TFTP-Server-Address
--------------    ----------------    ---------------------
boot.txt          *                   10.0.0.10
```

Showing DHCP server configuration settings for VRF **primary-vrf**.

```
switch# show dhcp-server vrf primary-vrf

VRF Name           : primary-vrf
DHCP Server        : disabled
Operational State  : disabled
Authoritative Mode : false

Pool Name          : test
Lease Duration     : 00:01:00

DHCP dynamic IP allocation
--------------------------
Start-IP-Address   End-IP-Address   Prefix-Length
----------------   --------------   --------------
10.0.0.1           10.0.0.30        *
192.168.1.1        192.168.1.20     24
192.168.10.30      192.168.10.60    16


DHCP Server options
-------------------
Option-Number   Option-Type    Option-Value
-------------   -----------    ------------
6               ip             10.0.0.3 10.0.0.4 10.0.0.5 10.0.0.6
18              ascii          aswed


DHCP Server static IP allocation
--------------------------------
IP-Address    Client-Hostname   MAC-Address
----------    ---------------   -----------------
10.0.0.1        *               aa:bb:cc:11:12:a4
20.0.0.1        *               11:22:11:22:aa:dd


BOOTP Options
---------------
Boot-File-Name    TFTP-Server-Name     State            TFTP-Server-Address
```

```
--------------    ----------------    ------          ---------------------
boot.txt          *                   OPERATIONAL     10.0.0.10
```

static-bind

Syntax

```
static-bind ip <IPV4-ADDR> mac <MAC-ADDR> [hostname <HOST>]

no static-bind <IPV4-ADDR-LIST>
```

Description

Creates a static binding that associates an IP address in the current pool with a specific MAC address. This causes the DHCPv4 server to only assign the specified IP address to a client station with the specified MAC address.

The `no` form of this command removes the specified binding.

Command context

```
config-dhcp-server-pool
```

Parameters

<IPV4-ADDR>

Specifies an IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255. The IP address must be within the address range defined for the current pool.

mac *<MAC-ADDR>*

Specifies a client station MAC address (`xx:xx:xx:xx:xx:xx`), where `x` is a hexadecimal number from 0 to F.

hostname *<HOST>*

Specifies the host name of the client station. Range: 1 to 255 printable ASCII characters

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines a static address for the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# static-bind ip 10.0.0.1 mac 24:be:05:24:75:73
```

Deletes a static address from the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcp-server)# pool primary-pool
switch(config-dhcp-server-pool)# no static-bind ip 10.0.0.1 mac 24:be:05:24:75:73
```

DHCP server IPv6 commands

authoritative

Syntax

```
authoritative

no authoritative
```

Description

Configures the DHCPv6 server as *authoritative* on the current VRF. This means that the server is the sole authority for the network on the VRF. It responds to client solicit messages with advertise messages having a priority/preference value set to 255 (the maximum), instead of 0 (the minimum). Clients always choose the DHCPv6 server with the highest priority/preference value. If two DHCPv6 servers send an advertise message with the same priority/preference value, then the client picks one and discards the other.

The `no` form of this command disables authoritative mode on the current VRF.

Command context

```
config-dhcpv6-server
```

Authority

Administrators or local user group members with execution rights for this command.

Example

Configures DHCPv6 server authoritative mode on VRF **`primary`**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# authoritative
```

Removes DHCPv6 server authoritative mode on VRF **`primary`**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# no authoritative
```

clear dhcpv6-server leases

Syntax

```
clear dhcpv6-server leases [all-vrfs | <IPV6-ADDR> vrf <VRF-NAME>] | vrf <VRF-NAME>]
```

Description

Clears DHCPv6 server lease information. The DHCPv6 server must be disabled before clearing lease information.

Command context

Manager (#)

Parameters

`all-vrfs`

Clears leases for all VRFs.

<IPV6-ADDR> vrf <VRF-NAME>

Clears the lease for a specific client on a specific VRF. Specify the client address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F. You can use two colons (::) to represent consecutive zeros (but only once), remove leading zeros, and collapse a hextet of four zeros to a single 0. For example, this address `2222:0000:3333:0000:0000:0000:4444:0055` becomes `2222:0:3333::4444:55`.

vrf <VRF-NAME>

Clears leases for a specific VRF.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Clearing all DHCPv6 server leases.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# disable
switch(config-dhcpv6-server)# exit
switch(config)# exit
switch# clear dhcpv6-server leases
```

Clearing all DHCPv6 server leases for VRF **primary-vrf**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# disable
switch(config-dhcpv6-server)# exit
switch(config)# exit
switch# clear dhcpv6-server leases vrf primary-vrf
```

Clear the DHCPv6 server lease for IP address **2001::1** on VRF **primary-vrf**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# disable
switch(config-dhcpv6-server)# exit
switch(config)# exit
switch# clear dhcpv6-server leases 2001::1 vrf primary-vrf
```

dhcv6p-server external-storage

Syntax

```
dhcpv6-server external-storage <VOLUME-NAME> file <LEASE-FILENAME> [delay <DELAY>]

no dhcpv6-server external-storage <VOLUME-NAME> file <LEASE-FILENAME> [delay <DELAY>]
```

Description

Configures the external storage file location for DHCPv6 server lease information. This file provides persistent storage, enabling DHCPv6 server settings to be restored when the switch is restarted. Lease information is stored in a flat file on the configured external device.

If external storage is not configured, then after a failure or reboot, all existing lease information is lost.

Lease information is saved to external storage each time the delay timer expires, which by default is every 300 seconds.

Lease information is not restored when issuing the command `dhcp-server enable`.

The `no` form of this command removes external storage support for the DHCPv6 server.

Command context

`config`

Parameters

<VOLUME-NAME>

Specifies the external storage volume name. Range: 1 to 64 printable ASCII characters.

`file` ***<LEASE-FILENAME>***

Specifies the external storage filename. Range: 1 to 255 printable ASCII characters.

`delay` ***<DELAY>***

Specifies the interval in seconds between updates to the external storage file. Range: 15 to 86400. Default: 300.

Authority

Administrators or local user group members with execution rights for this command.

Example

Stores the lease file on external storage volume **`Storage1`** in file **`LeaseFile`** at an interval of 600 seconds.

```
switch(config)# dhcpv6-server external-storage Storage1 file LeaseFile delay 600
```

Disables storage of the lease file on external storage volume **`Storage1`** in file **`LeaseFile`**.

```
switch(config)# no dhcpv6-server external-storage Storage1 file LeaseFile delay 600
```

dhcpv6-server vrf

Syntax

`dhcpv6-server vrf` *`VRF-NAME`*

`no dhcpv6-server vrf` *`VRF-NAME`*

Description

Configures the DHCPv6 server to support a VRF and changes to the `config-dhcpv6-server` context for that VRF.

The `no` form of this command removes DHCPv6 server support on a VRF.

Command context

`config`

Parameters

VRF-NAME

Name of a VRF.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configures DHCPv6 server support on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
```

Removes the DHCPv6 server support on VRF **primary**.

```
switch(config)# no dhcpv6-server vrf primary
```

disable

Syntax

```
disable
```

Description

Disables the DHCPv6 server on the current VRF. The DHCPv6 server is disabled by default when configured on a VRF.

Command context

```
config-dhcpv6-server
```

Authority

Administrators or local user group members with execution rights for this command.

Example

Disables the DHCPv6 server on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# disable
```

dns-server

Syntax

```
dns-server <IPVv6-ADDR-LIST>

no dns-server <IPVv6-ADDR-LIST>
```

Description

Defines up to four DNS servers for the current DHCPv6 server pool.

The `no` form of this command removes the specified DNS servers from the pool.

Command context

```
config-dhcpv6-server-pool
```

Parameters

<IPVv6-ADDR-LIST>

Specifies the IP addresses of the DNS servers in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F. Separate addresses with a space. A maximum of four IP addresses can be defined.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines DNS server **2001::13** for the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# dns-server 2001::13
```

Deletes DNS server **2001::13** from the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# no dns-server 2001::13
```

enable

Syntax

```
enable
```

Description

Enables the DHCPv6 server on the current VRF. The DHCPv6 server is disabled by default when configured on a VRF.

Command context

```
config-dhcpv6-server
```

Authority

Administrators or local user group members with execution rights for this command.

Example

Enables the DHCPv6 server on VRF **primary**.

```
switch(config)# dhcp-server vrf primary
switch(config-dhcpv6-server)# enable
```

lease

Syntax

```
lease {<TIME> | infinite}

no lease
```

Description

Sets the length of the DHCPv6 lease time for the current pool. The lease time determines how long an IP address is valid before a DHCPv6 client must request that it be renewed.

The `no` form of this command returns the DHCPv6 lease time to the default value 1 hour.

Command context

```
config-dhcpv6-server-pool
```

Parameters

<TIME>

Sets the DHCPv6 lease time. Format: DD:HH:MM. Default: 01:00:00.

infinite

Sets the DHCPv6 lease time to infinite. This means that addresses do not need to be renewed.

Authority

Administrators or local user group members with execution rights for this command.

Example

Sets the lease time for DHCPv6 server pool **primary-pool** on VRF **primary** to **12** hours.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# lease 00:12:00
```

Sets the lease time for DHCP server pool **primary-pool** on VRF **primary** to the default value.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# no lease 00:12:00
```

option

Syntax

```
option <OPTION-NUM> {ascii <ASCII-STR> | hex <HEX-STR> | ip <IPV6-ADDR-LIST>}

no option <OPTION-NUM> {ascii <ASCII-STR> | hex <HEX-STR> | ip <IPV6-ADDR-LIST>}
```

Description

Defines custom DHCPv6 options for the current DHCPv6 server pool.

The `no` form of this command removes custom DHCPv6 options from the pool.

Command context

```
config-dhcpv6-server-pool
```

Parameters

<OPTION-NUM>

Specifies a DHCPv6 option number. Range: 2 to 254.

ascii *<ASCII-STR>*

Specifies a value for the selected option as an ASCII string. Range: 1 to 255 ASCII characters.

hex *<HEX-STR>*

Specifies a value for the selected option as a hexadecimal string. Range: 1 to 255 hexadecimal characters.

ip *<IPV6-ADDR-LIST>*

Specifies a list of IP addresses for the option in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines DHCPv6 option **22** for the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# option 22 ipv6 2001::12
```

Deletes DHCPv6 option 22 for the server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# no option 22 ipv6 2001::12
```

pool

Syntax

```
pool <POOL-NAME>

no pool <POOL-NAME>
```

Description

Creates a DHCPv6 server pool for the current VRF and switches to the `config-dhcpv6-server-pool` context for it. Multiple pools, each with a distinct range, can be assigned to a VRF. A maximum of 64 pools (IPv4 and IPv6), 64 address ranges, and 8182 clients are supported on the switch across all VRFs.

The `no` form of this command deletes the specified DHCPv6 server pool.

Command context

`config-dhcpv6-server`

Parameters

<POOL-NAME>

Specifies the DHCPv6 pool name. A maximum of 64 pools (IPv4 and IPv6) are supported across VRFs on the switch. Range: 1 to 32 printable ASCII characters. First character must be a letter or number.

Authority

Administrators or local user group members with execution rights for this command.

Example

Creates the DHCPv6 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)#
```

Deletes the DHCPv6 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# no pool primary-pool
```

range

Syntax

```
range <LOW-IPV6-ADDR> <HIGH-IPV6-ADDR> [prefix-len <MASK>]

no range <LOW-IPV6-ADDR> <HIGH-IPV6-ADDR> [prefix-len <MASK>]
```

Description

Defines the range of IP addresses supported by the current DHCPv6 server pool. A maximum of 64 ranges are supported per switch across all VRFs.

The `no` form of this command deletes the address range for the current pool.

Command context

```
config-dhcpv6-server-pool
```

Parameters

<LOW-IPV6-ADDR>

Specifies the lowest IP address in the pool in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

<HIGH-IPV6-ADDR>

Specifies the highest IP address in the pool in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

prefix-len *<MASK>*

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 64 to128.

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines an address range for the DHCPv6 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# range 2001::1 2001::10 prefix-len 64
```

Deletes an address range for the DHCPv6 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# no range 2001::1 2001::10 prefix-len 64
```

show dhcpv6-server

Syntax

```
show dhcpv6-server [all-vrfs]

show dhcpv6-server leases {all-vrfs | vrf <VRF-NAME>}

show dhcpv6-server pool <POOL-NAME> [vrf <VRF-NAME>]
```

Description

Shows configuration settings for the DHCPv6 server.

Command context

Manager (#)

Parameters

`all-vrfs`

Shows DHCPv6 server configuration settings for all VRFs.

`leases {all-vrfs | vrf <VRF-NAME>}`

Shows DHCPv6 server lease configuration settings for all VRFs or a specific VRF.

`pool <POOL-NAME> [vrf <VRF-NAME>]`

Shows DHCPv6 server pool configuration settings for all VRFs or a specific VRF.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing all DHCPv6 server configuration settings.

```
switch# show dhcpv6-server

VRF Name           : default
DHCPv6 Server      : enabled
Operational State  : operational
Authoritative Mode : true

Pool Name          : test
Lease Duration     : 00:01:00

DHCPV6 dynamic IP allocation
----------------------------
Start-IPv6-Address  End-IPv6-Address   Prefix-Length
------------------  ----------------   -------------
2001::2             2001::10           64

DHCPv6 Server options
---------------------
Option-Number     Option-Type    Option-Value
-------------     -----------    ------------
7                 ipv6           2001::15

DHCvP6 Server static IP allocation
----------------------------------
DHCPv6 Server static host is not configured.
```

Showing DHCPv6 server configuration settings for VRF **`primary-vrf`**.

```
switch# show dhcpv6-server vrf primary-vrf

VRF Name           : primary-vrf
DHCPv6 Server      : disabled
Operational State  : standby
Authoritative Mode : false

Pool Name          : test
Lease Duration     : 00:01:00
```

```
DHCPV6 dynamic IP allocation
----------------------------
Start-IPv6-Address  End-IPv6-Address   Prefix-Length
------------------  ----------------  -------------
2000::1             2000::20          *
2001::20            2001::50          *
2001::2             2001::10          64
2010::20            2010::40          *


DHCPv6 Server options
---------------------
Option-Number     Option-Type    Option-Value
-------------     -----------    ------------
7                 ipv6           2001::15
23                ipv6           2001::30
30                ipv6           2001::10


DHCvP6 Server static IP allocation
----------------------------------
DHCPv6 Server static host is not configured.

Pool Name       : v6test
Lease Duration  : 00:01:00

DHCPv6 dynamic IP allocation
----------------------------
Start-IPv6-Address    End-IPv6-Address    Prefix-Length
------------------    ----------------    -------------
2001::1               2001::20            64
2010::10              2010::30            *
2020::20              2020::60            *


DHCPv6 Server options
---------------------
Option-Number    Option-Type    Option-Value
-------------    -----------    -----------------
7                ipv6           2001::20
23               ipv6           2001:0db8:85a3:0000:0000:8a2e:0370:7334 2001:0db8:85a3:0000:0000:8a2e:0370:7335
                                2001:0db8:85a3:0000:0000:8a2e:0370:7336 2001:0db8:85a3:0000:0000:8a2e:0370:7337


DHCPv6 Server static IP allocation
----------------------------------
IPv6-Address    Client-Hostname     State         Client-Id
------------    ---------------     -----------   ---------
2100::4         *                   OPERATIONAL   1:0:a0:24:ab:fb:9c
```

static-bind

Syntax

`static-bind ipv6 <IPVv6-ADDR> client-id <ID> [hostname <HOST>]`

`no static-bind ipv6 <IPVv6-ADDR-LIST>`

Description

Creates a static binding that associates an IP address in the current pool with a client identifier or DUID. This causes the DHCPv6 server to only assign the specified IP address to a client station with the specified client identifier or DUID.

The `no` form of this command removes the specified static binding from the pool.

Command context

`config-dhcpv6-server-pool`

Parameters

<IPV6-ADDR>

Specifies the IP address to assign in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F. For example, this address `2222:0000:3333:0000:0000:0000:4444:0055` becomes `2222:0:3333::4444:55`.

client-id *<ID>*

Specifies the client identifier or DUID.

hostname *<HOST>*

Specifies the host name of the client station. Range: 1 to 255 printable ASCII characters

Authority

Administrators or local user group members with execution rights for this command.

Example

Defines a static address for the DHCPv6 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# static-bind ipv6 2001::10 client-id 1:0:a0:24:ab:fb:9c
```

Deletes a static address from the DHCPv6 server pool **primary-pool** on VRF **primary**.

```
switch(config)# dhcpv6-server vrf primary
switch(config-dhcpv6-server)# pool primary-pool
switch(config-dhcpv6-server-pool)# no static-bind ipv6 2001::10 client-id 1:0:a0:24:ab:fb:9c
```

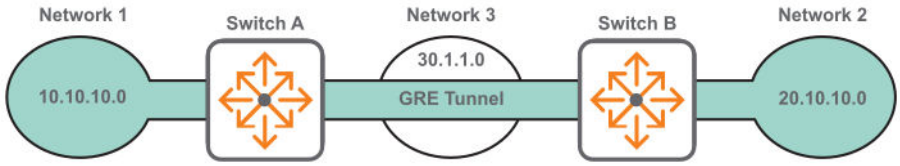
True point-to-point networks are not always possible in corporate networking environment. Many networks deploy nontraditional methods of connection (for example, DSL or broadband) at remote sites or branch offices. The branch office, telecommuter, or business traveler then becomes separated from the corporate network. Some method of tunneling becomes imperative to connect all the network sites together.

Virtual Private Networking (VPN) is often deployed to create private tunnels through the public network system for passing data to remote sites. While VPN is sufficient for the average business traveler, it is not a good solution for branch site connectivity. VPN configurations must include statically maintained access lists to identify traffic through the tunnel. These access lists are often tedious to configure for larger networks and are prone to errors.

VPNs do not permit multicast traffic to pass; therefore routing protocols such as Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) are no longer options for dynamic routing updates. All new additions to the network topology must be manually added to the various configured access lists. Without dynamic routing from one site to another, network management is severely hampered. Network managers need their non-heterogeneous networks to function like traditional point-to-point networks so that traditional management methods (once available only on point-to-point circuits) can apply to the entire network.

The solution to these challenges is to use IP tunnels. An IP tunnel provides a virtual link between endpoints on two different networks enabling data to be exchanged as if the endpoints were directly connected on the same network. Traffic between the devices is isolated from the intervening networks that the tunnel spans.

For example, the following diagram shows an IP tunnel (using GRE) that connects two IPv4 networks over an IPv4 network.



If network 1 and network 3 are using IPv6 addressing, the tunnel connects them by encapsulating the IPv6 traffic in IPv4 packets to traverse network 2. The intermediate network devices do not know about Network 1 and Network 2 because the packets are encapsulated.

An IP tunnel can also be used to create a point-to-point link for IPv6 traffic over an IPv6 network.

IP tunnels supported features

- Up to 127 tunnels can be defined on a switch shared between different tunnel types: GRE, IPv6 in IPv4, and IPv6 in IPv6.
- A maximum of 16 source IP addresses are supported. Tunnels can have the same source IP address and different destination IP addresses. The source IP, destination IP, and VRF combine to uniquely identify a tunnel.

Unsupported features

- GRE IPv4 over IPv6.
- QoS cannot be applied to a GRE tunnel interface.
- Key support can be added for security and identification purposes when there are multiple applications.

- VPN across public IP network.
- MPLS over GRE.
- Multipoint GRE for scalable network to reach multiple remote sites.

Configuring an IP tunnel

Prerequisites

An enabled layer 3 interface with an IP address assigned to it, created with the command `interface`.

Procedure

1. Create an IP tunnel with the command **`interface tunnel`** .
2. Set the IP address for the tunnel. For a GRE tunnel, enter the command **`ip address ip address`**. For an IPv6 in IPv4 or an IPv6 in IPv6 tunnel, enter the command **`ipv6 address`**.
3. Set the source IP address for the tunnel. For a GRE or an IPv6 in IPv4 tunnel, enter the command **`source ip`**. For an IPv6 in IPv6 tunnel, enter the command **`source ipv6`**.
4. Set the destination IP address for the tunnel. For a GRE or an IPv6 in IPv4 tunnel, enter the command **`destination ip`** . For an IPv6 in IPv6 tunnel, enter the command **`destination ipv6`**.
5. Optionally, set the TTL (hop count) for the tunnel with the command **`ttl`**.
6. Optionally, set the MTU for the tunnel with the command **`ip mtu`**.
7. Optionally, add a description to the tunnel with the command **`description`**.
8. By default, the tunnel is attached to the default VRF. Attach it to a different VRF with the command **`vrf attach`**.
9. Enable the tunnel with the command **`no shutdown`**.
10. Review tunnel settings with the command **`show interface tunnel`**.

Example

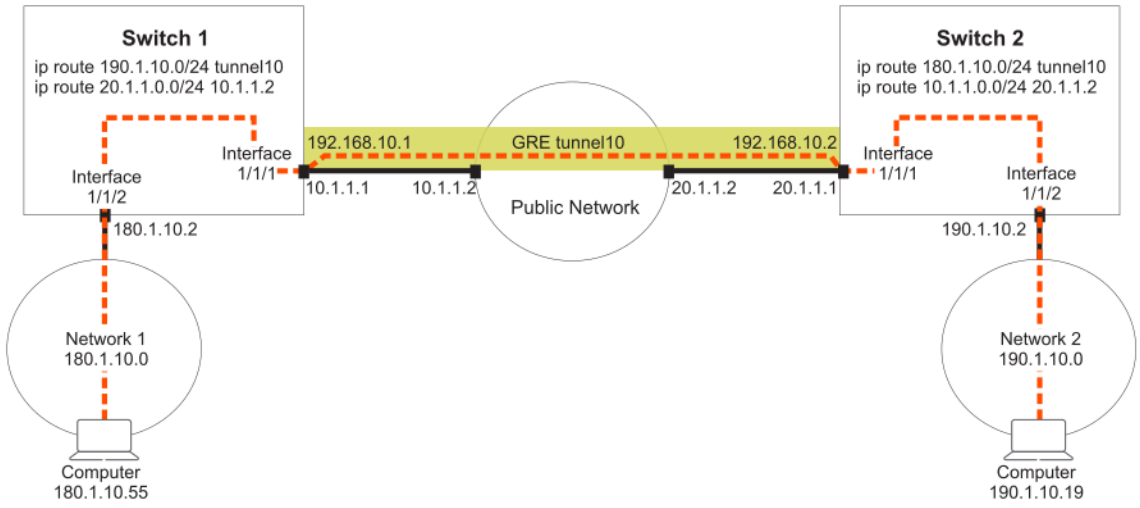
This example creates the following configuration:

- Creates GRE tunnel **`33`**.
- Set the tunnel IP address to **`10.10.20.209/24`**.
- Sets the tunnel source IP address to **`10.10.10.1`**.
- Sets the tunnel destination IP address to **`10.10.10.2`**.
- Enables the tunnel.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# ip address 10.10.20.209/24
switch(config-gre-if)# source ip address 10.10.10.1
switch(config-gre-if)# destination ip address 10.10.10.2
switch(config-gre-if)# no shutdown
```

Creating a GRE tunnel for traversing a public network

This example creates a GRE tunnel between two switches, enabling traffic from two networks to traverse a public network.



Procedure

1. On switch 1:

   a. Enable interface **1/1/1** and assign the IP address **10.1.1.1/24** to it.

   ```
   switch# config
   switch(config)# interface 1/1/1
   switch(config-if)# ip address 10.1.1.1/24
   switch(config-if)# no shutdown
   ```

   b. Enable interface **1/1/2** and assign the IP address **180.1.10.2/24** to it.

   ```
   switch# config
   switch(config)# interface 1/1/2
   switch(config-if)# ip address 180.1.10.2/24
   switch(config-if)# no shutdown
   switch(config-if)# exit
   ```

   c. Create GRE tunnel **10** and assign the IP address **192.168.10.1/24**, source address **10.1.1.1**, and destination address **20.1.1.1** to it.

   ```
   switch(config)# interface tunnel 10 mode gre ipv4
   switch(config-gre-if)# ip address 192.168.10.1/24
   switch(config-gre-if)# source ip 10.1.1.1
   switch(config-gre-if)# destination ip 20.1.1.1
   switch(config-gre-if)# no shutdown
   switch(config-gre-if)# exit
   ```

   d. Defines routes so that traffic from network 1 can reach network 2 through the tunnel.

   ```
   switch(config)# ip route 20.1.1.0/24 10.1.1.2
   switch(config)# ip route 190.1.10.0/24 tunnel10
   ```

2. On switch 2:

a. Enable interface **1/1/1** and assign the IP address **20.1.1.1/24** to it.

```
switch# config
switch(config)# interface 1/1/1
switch(config-if)# ip address 20.1.1.1/24
switch(config-if)# no shutdown
```

b. Enable interface **1/1/2** and assign the IP address **190.1.10.2/24** to it.

```
switch(config)# interface 1/1/2
switch(config-if)# ip address 190.1.10.2/24
switch(config-if)# no shutdown
switch(config-if)# exit
```

c. Create GRE tunnel **10** and assign the IP address **192.168.10.2/24**, source address **20.1.1.1**, and destination address **10.1.1.1** to it.

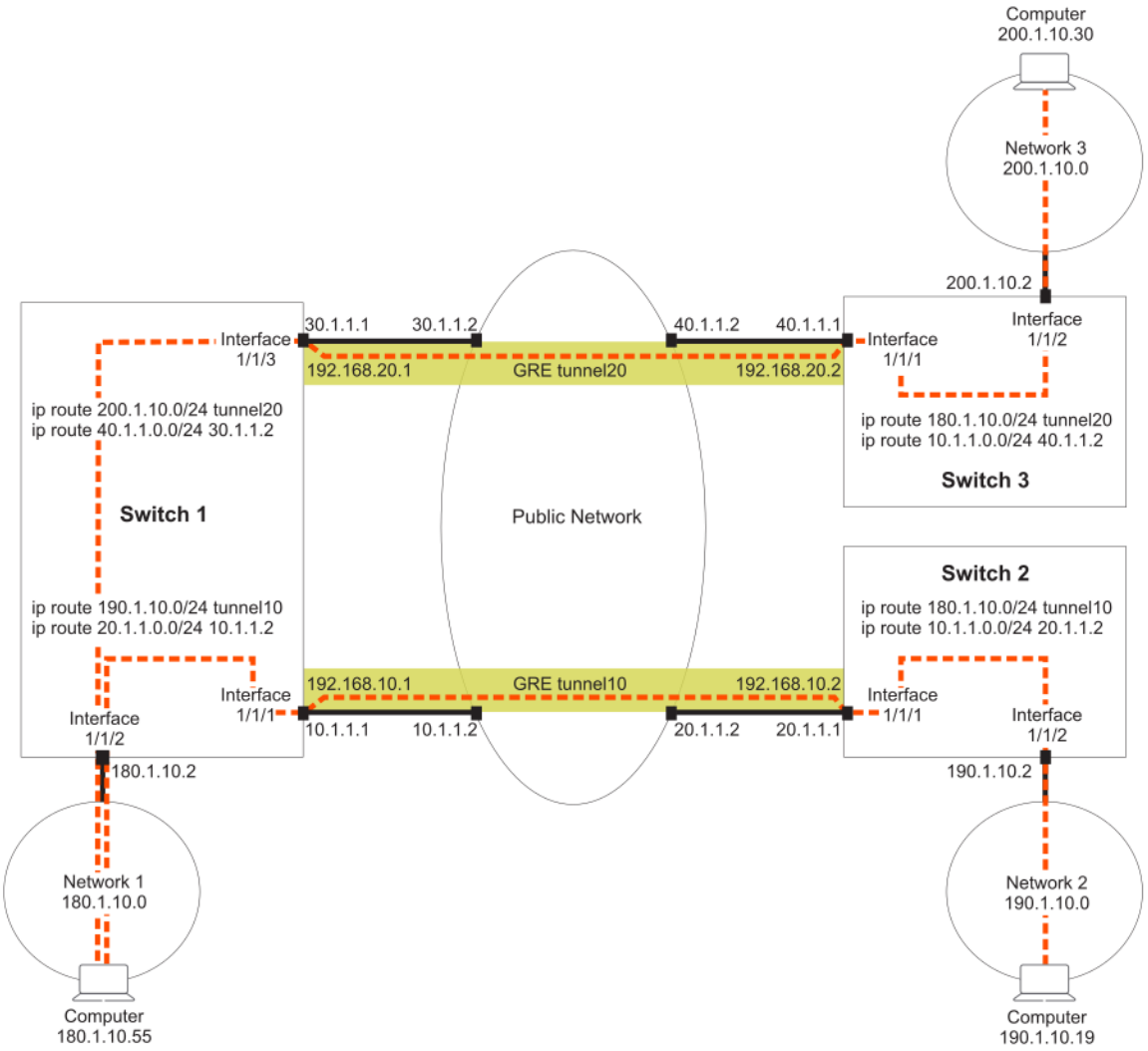
```
switch(config)# interface tunnel 10 mode gre ipv4
switch(config-gre-if)# ip address 192.168.10.2/24
switch(config-gre-if)# source ip 20.1.1.1
switch(config-gre-if)# destination ip 10.1.1.1
switch(config-gre-if)# no shutdown
switch(config-gre-if)# exit
```

d. Defines routes so that traffic from network 2 can reach network 1 through the tunnel.

```
switch(config)# ip route 10.1.1.0/24 20.1.1.2
switch(config)# ip route 180.1.10.0/24 tunnel10
```

Creating two GRE tunnels to different destination addresses

This example creates two GRE tunnels to different destination addresses. Traffic from network 1 can reach either network 2 or network 3 using the appropriate tunnel.



Procedure

1. On switch 1:

   a. Enable interface **1/1/1** and assign the IP address **10.1.1.1/24** to it.

   ```
   switch# config
   switch(config)# interface 1/1/1
   switch(config-if)# ip address 10.1.1.1/24
   switch(config-if)# no shutdown
   ```

   b. Enable interface **1/1/2** and assign the IP address **180.1.10.2/24** to it.

   ```
   switch# config
   switch(config)# interface 1/1/2
   switch(config-if)# ip address 180.1.10.2/24
   switch(config-if)# no shutdown
   switch(config-if)# exit
   ```

c. Enable interface **1/1/3** and assign the IP address **30.1.1.1/24** to it.

```
switch# config
switch(config)# interface 1/1/3
switch(config-if)# 30.1.1.1/24
switch(config-if)# no shutdown
switch(config-if)# exit
```

d. Create GRE tunnel **10** and assign the IP address **192.168.10.1/24**, source address **10.1.1.1**, and destination address **20.1.1.1** to it.

```
switch(config)# interface tunnel 10 mode gre ipv4
switch(config-gre-if)# ip address 192.168.10.1/24
switch(config-gre-if)# source ip 10.1.1.1
switch(config-gre-if)# destination ip 20.1.1.1
switch(config-gre-if)# no shutdown
switch(config-gre-if)# exit
```

e. Create GRE tunnel **20** and assign the IP address **192.168.20.1/24**, source address **30.1.1.1**, and destination address **40.1.1.1** to it.

```
switch(config)# interface tunnel 20 mode gre ipv4
switch(config-gre-if)# ip address 192.168.20.1/24
switch(config-gre-if)# source ip 30.1.1.1
switch(config-gre-if)# destination ip 40.1.1.1
switch(config-gre-if)# no shutdown
switch(config-gre-if)# exit
```

f. Defines routes so that traffic from network 1 can reach network 2 through tunnel 10.

```
switch(config)# ip route 20.1.1.0/24 10.1.1.2
switch(config)# ip route 190.1.10.0/24 tunnel10
```

g. Defines routes so that traffic from network 1 can reach network 3 through the tunnel 20.

```
switch(config)# ip route 40.1.1.0/24 30.1.1.2
switch(config)# ip route 200.1.10.0/24 tunnel20
```

2. On switch 2:

a. Enable interface **1/1/1** and assign the IP address **20.1.1.1/24** to it.

```
switch# config
switch(config)# interface 1/1/1
switch(config-if)# ip address 20.1.1.1/24
switch(config-if)# no shutdown
```

b. Enable interface **1/1/2** and assign the IP address **190.1.10.2/24** to it.

```
switch(config)# interface 1/1/2
switch(config-if)# ip address 190.1.10.2/24
switch(config-if)# no shutdown
switch(config-if)# exit
```

c. Create GRE tunnel **10** and assign the IP address **192.168.10.2/24**, source address **20.1.1.1**, and destination address **10.1.1.1** to it.

```
switch(config)# interface tunnel 10 mode gre ipv4
switch(config-gre-if)# ip address 192.168.10.2/24
switch(config-gre-if)# source ip 20.1.1.1
```

```
switch(config-gre-if)# destination ip 10.1.1.1
switch(config-gre-if)# no shutdown
switch(config-gre-if)# exit
```

d. Defines routes so that traffic from network 2 can reach network 1 through tunnel 10.

```
switch(config)# ip route 10.1.1.0/24 20.1.1.2
switch(config)# ip route 180.1.10.0/24 tunnel10
```

3. On switch 3:

a. Enable interface **1/1/1** and assign the IP address **40.1.1.1/24** to it.

```
switch# config
switch(config)# interface 1/1/1
switch(config-if)# ip address 40.1.1.1/24
switch(config-if)# no shutdown
```

b. Enable interface **1/1/2** and assign the IP address **200.1.10.2/24** to it.

```
switch(config)# interface 1/1/2
switch(config-if)# ip address 200.1.10.2/24
switch(config-if)# no shutdown
switch(config-if)# exit
```

c. Create GRE tunnel **20** and assign the IP address **192.168.20.2/24**, source address **40.1.1.1**, and destination address **30.1.1.1** to it.

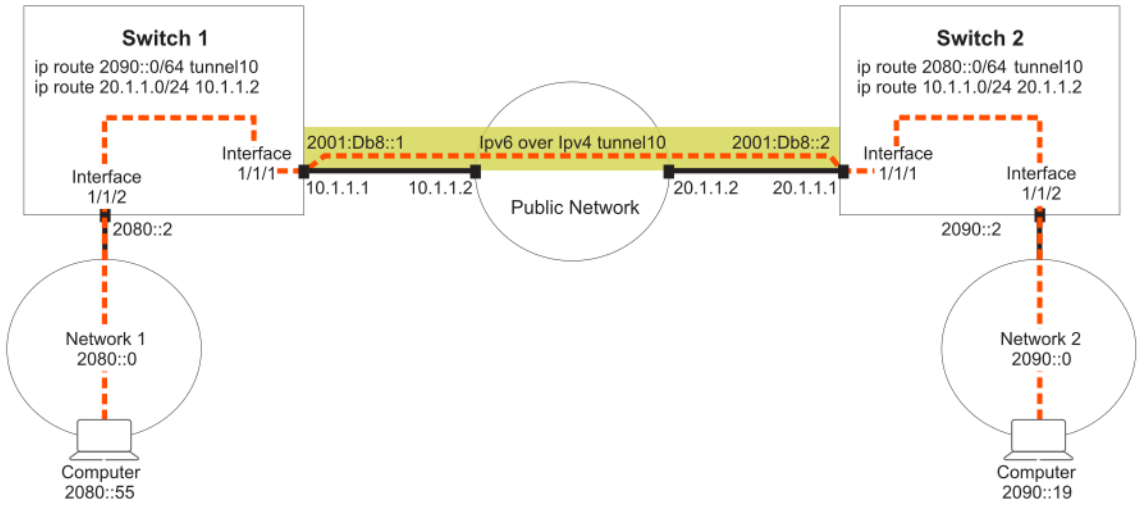
```
switch(config)# interface tunnel 10 mode gre ipv4
switch(config-gre-if)# ip address 192.168.20.2/24
switch(config-gre-if)# source ip 40.1.1.1
switch(config-gre-if)# destination ip 30.1.1.1
switch(config-gre-if)# no shutdown
switch(config-gre-if)# exit
```

d. Defines routes so that traffic from network 3 can reach network 1 through tunnel 20.

```
switch(config)# ip route 30.1.1.0/24 40.1.1.2
switch(config)# ip route 180.1.10.0/24 tunnel20
```

Creating an IPv6 in IPv4 tunnel for traversing a public network

This example creates an IPv6 in IPv4 tunnel between two switches, enabling traffic from two networks to traverse a public network.



Procedure

1. On switch 1:

   a. Enable interface `1/1/1` and assign the IP address `10.1.1.1/24` to it.

   ```
   switch# config
   switch(config)# interface 1/1/1
   switch(config-if)# ip address 10.1.1.1/24
   switch(config-if)# no shutdown
   ```

   b. Enable interface `1/1/2` and assign the IP address `2080::2/64` to it.

   ```
   switch# config
   switch(config)# interface 1/1/2
   switch(config-if)# ipv6 address 2080::2/64
   switch(config-if)# no shutdown
   switch(config-if)# exit
   ```

   c. Create IPv6 in IPv4 tunnel `10` and assign the IP address `2001:DB8::1/32`, source address `10.1.1.1`, and destination address `20.1.1.1` to it.

   ```
   switch(config)# interface tunnel 10 mode ip 6in4
   switch(config-ip-if)# ipv6 address 2001:DB8::1/62
   switch(config-ip-if)# source ip 10.1.1.1
   switch(config-ip-if)# destination ip 20.1.1.1
   switch(config-ip-if)# no shutdown
   switch(config-ip-if)# exit
   ```

   d. Defines routes so that traffic from network 1 can reach network 2 through the tunnel.

   ```
   switch(config)# ip route 20.1.1.0/24 10.1.1.2
   switch(config)# ipv6 route 290::0/64 tunnel10
   ```

2. On switch 2:

   a. Enable interface `1/1/1` and assign the IP address `20.1.1.1/24` to it.

   ```
   switch# config
   switch(config)# interface 1/1/1
   ```

```
switch(config-if)# ip address 20.1.1.1/24
switch(config-if)# no shutdown
```

b. Enable interface **1/1/2** and assign the IP address **2090::2/64** to it.

```
switch(config)# interface 1/1/2
switch(config-if)# ipv6 address 2090::2/64
switch(config-if)# no shutdown
switch(config-if)# exit
```

c. Create IPv6 in IPv4 tunnel **10** and assign the IP address **2001:DB8::2/32**, source address **10.1.1.1**, and destination address **20.1.1.1** to it.

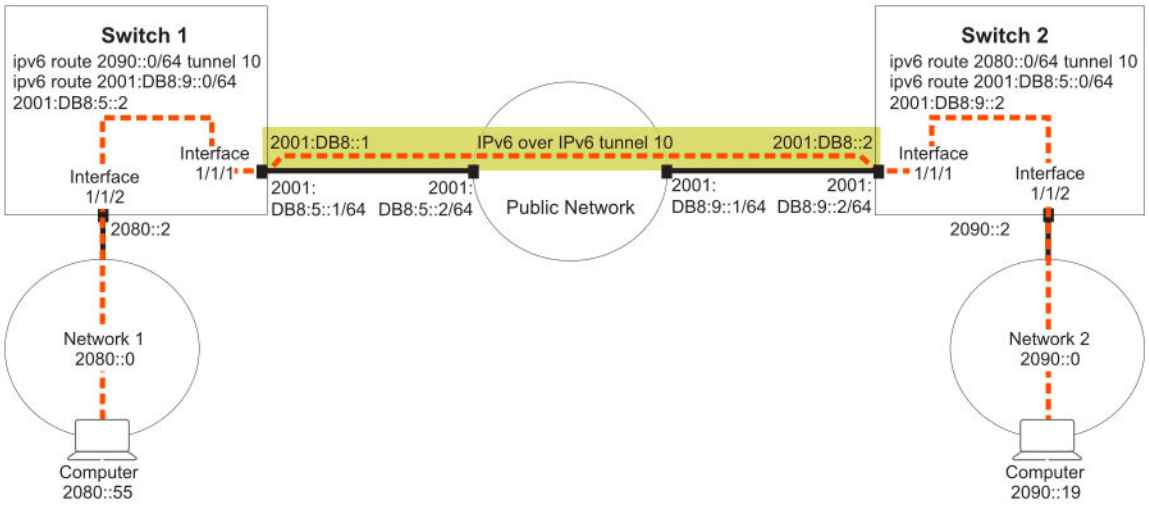
```
switch(config)# interface tunnel 10 mode ip 6in4
switch(config-ip-if)# ipv6 address 2001:DB8::2/62
switch(config-ip-if)# source ip 20.1.1.1
switch(config-ip-if)# destination ip 10.1.1.1
switch(config-ip-if)# no shutdown
switch(config-ip-if)# exit
```

d. Defines routes so that traffic from network 2 can reach network 1 through the tunnel.

```
switch(config)# ip route 10.1.1.0/24 20.1.1.2
switch(config)# ip route 2080::0/64 tunnel10
```

Creating an IPv6 in IPv6 tunnel for traversing a public network

This example creates an IPv6 in IPv6 tunnel between two switches, enabling traffic from two networks to traverse a public network.



Procedure

1. On switch 1:

a. Enable interface **1/1/1** and assign the IP address **2001:DB8:5::1/64** to it.

```
switch# config
switch(config)# interface 1/1/1

switch(config-if)# ipv6 address 2001:DB8:5::1/64
switch(config-if)# no shutdown
```

b. Enable interface **1/1/2** and assign the IP address **2080::2/64** to it.

```
switch# config
switch(config)# interface 1/1/2
switch(config-if)# ipv6 address 2080::2/64
switch(config-if)# no shutdown
switch(config-if)# exit
```

c. Create IPv6 in IPv6 tunnel **10** and assign the IP address **2001:DB8::1/32**, source address **2001:DB8:5::1**, and destination address **2001:DB8:9::1** to it. (Optional) Set the MTU and TTL parameters for this tunnel interface.

```
switch(config)# interface tunnel 10 mode ip 6in6
switch(config-ip-if)# ipv6 address 2001:DB8::1/62
switch(config-ip-if)# source ipv6 2001:DB8:5::1
switch(config-ip-if)# destination ipv6 2001:DB8:9::1
switch(config-ip-if)# no shutdown
switch(config-ip-if)# exit
```

d. Defines routes so that traffic from network 1 can reach network 2 through the tunnel.

```
switch(config)# ipv6 route 2001:DB8:9::0/64 2001:DB8:5::2
switch(config)# ipv6 route 2090::0/64 tunnel10
```

2. On switch 2:

a. Enable interface **1/1/1** and assign the IP address **2001:DB8:9::1/64** to it.

```
switch# config
switch(config)# interface 1/1/1
switch(config-if)# ipv6 address 2001:DB8:9::1/64
switch(config-if)# no shutdown
```

b. Enable interface **1/1/2** and assign the IP address **2090::2/64** to it.

```
switch(config)# interface 1/1/2
switch(config-if)# ipv6 address 2090::2/64
switch(config-if)# no shutdown
switch(config-if)# exit
```

c. Create IPv6 in IPv6 tunnel **10** and assign the IP address **2001:DB8::2/32**, source address **2001:DB8:5::1**, and destination address **2001:DB8:9::1** to it. (Optional) Set the MTU and TTL parameters for this tunnel interface.

```
switch(config)# interface tunnel 10 mode ip 6in6
switch(config-ip-if)# ipv6 address 2001:DB8::2/62
switch(config-ip-if)# source ipv6 2001:DB8:9::1
switch(config-ip-if)# destination ipv6 2001:DB8:5::1
```

```
switch(config-ip-if)# no shutdown
switch(config-ip-if)# exit
```

d. Defines routes so that traffic from network 2 can reach network 1 through the tunnel.

```
switch(config)# ipv6 route 2001:DB8:5::0/64 2001:DB8:9::2
switch(config)# ipv6 route 2080::0/64 tunnel10
```

IP tunnels commands

description

Syntax

```
description <DESC>

no description
```

Description

Associates a text description with an IP tunnel for identification purposes.

The `no` form of this command removes the description from an IP tunnel.

Command context

```
config-gre-if

config-ip-if
```

Parameters

<DESC>

Specifies the descriptive text to associate with the IP tunnel. Range: 1 to 64 printable ASCII characters.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defines a description for GRE tunnel **33**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# description Network A Tunnel C
```

Removes the description for GRE tunnel **33**.

```
switch(config)# interface tunnel 33
switch(config-gre-if)# no description
```

Defines a description for IPv6 in IPv4 tunnel **27**.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)# description Network 3 Tunnel 27
```

Removes the description for IPv6 in IPv4 tunnel **27**.

```
switch(config)# interface tunnel 27
switch(config-ip-if)# no description
```

Defines a description for IPv6 in IPv6 tunnel **8**.

```
switch(config)# interface tunnel 8 mode ip 6in6
switch(config-ip-if)# description Network 4 Tunnel 8
```

Removes the description for IPv6 in IPv6 tunnel **8**.

```
switch(config)# interface tunnel 8
switch(config-ip-if)# no description
```

destination ip

Syntax

```
destination ip <IPV4-ADDR>

no destination ip <IPV4-ADDR>
```

Description

Sets the destination IP address for an IP tunnel. Specify the address of the interface on the remote device to which the tunnel will be established.

The `no` form of this command deletes the destination IP address from an IP tunnel.

Command context

```
config-gre-if

config-ip-if
```

Parameters

<IPV4-ADDR>

Specifies the destination IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defines the destination IP address to be **10.10.10.1** for GRE tunnel **33**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# destination ip 10.10.10.1
```

Deletes the destination IP address **10.10.10.1** from GRE tunnel **33**.

```
switch(config)# interface tunnel 33
 switch(config-gre-if)# no destination ip 10.10.10.1
```

Defines the destination IP address to be **10.10.20.1** for IPv6 in IPv4 tunnel **27**.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)# destination ip 10.10.20.1
```

Deletes the destination IP address **10.10.20.1** from IPv6 in IPv4 tunnel **27**.

```
switch(config)# interface tunnel 27
 switch(config-ip-if)# no destination ip 10.10.20.1
```

destination ipv6

Syntax

```
destination ipv6 <IPVv6-ADDR>

no destination ipv6 <IPV6-ADDR>
```

Description

Sets the destination IPv6 address for an IP tunnel. Specify the address of the interface on the remote device to which the tunnel will be established.

The `no` form of this command deletes the destination IPv6 address from an IP tunnel.

Command context

```
config-ip-if
```

Parameters

<IPV6-ADDR>

Specifies the tunnel IP address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defines the destination IPv6 address to be **`2001:DB8::1`** for IPv6 in IPv6 tunnel .

```
switch(config)# interface tunnel 8 mode ip 6in6
switch(config-ip-if)# destination ipv6 2001:DB8::1
```

Deletes the destination IPv6 address **`2001:DB8::1`** from IPv6 in IPv6 tunnel **`8`**.

```
switch(config)# interface tunnel 8
 switch(config-ip-if)# no destination ipv6 2001:DB8::1
```

interface tunnel

Syntax

```
interface tunnel <TUNNEL-NUMBER> mode {gre ipv4 | ip 6in4 | ip 6in6}

interface tunnel <EXISTING-TUNNEL-NUMBER>

no interface tunnel <EXISTING-TUNNEL-NUMBER>
```

Description

Creates or updates an IP tunnel. After you enter the command, the firmware switches to the configuration context for the tunnel.

If the specified tunnel exists, this command switches to the context for the tunnel.

By default, all tunnels are automatically assigned to the default VRF when they are created.

The `no` form of this command deletes an existing IP tunnel.

Command context

`config`

Parameters

`mode {gre ipv4 | ip 6in4 | ip 6in6}`

Creates an IP tunnel. Choose one of the following options:

- `gre ipv4`: Creates a GRE tunnel.
- `ip 6in4`: Creates an IPv4 tunnel for IPv6 traffic.
- `ip 6in6`: Creates an IPv6 tunnel for IPv6 traffic.

<TUNNEL-NUMBER>

Specifies the number for a new tunnel. Range: 1 to 127. Numbering is shared between all tunnels, so the same tunnel number cannot be used for an IPv6 in IPv4 tunnel and a GRE tunnel.

<EXISTING-TUNNEL-NUMBER>

Specifies the number for an existing IP tunnel. Range: 1 to 127.

Command context

`config-gre-if`

`config-ip-if`

Examples

Defines a new GRE tunnel with number **27**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)#
```

Switches to the `config-gre-if` context for existing tunnel **33**.

```
switch(config)# interface tunnel 33
switch(config-gre-if)#
```

Deletes GRE tunnel **33**.

```
switch(config)# no interface tunnel 33
```

Defines a new IPv6 in IPv4 tunnel with number **27**.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)#
```

Switches to the `config-ip-if` context for existing tunnel **27**.

```
switch(config)# interface tunnel 27
switch(config-ip-if)#
```

DeletesIPv6 in IPv4 tunnel **27**.

```
switch(config)# no interface tunnel 27
```

Defines a new IPv6 in IPv6 tunnel with number **8**.

```
switch(config)# interface tunnel 8 mode ip 6in6
switch(config-ip-if)#
```

ip address

Syntax

```
ip address <IPV4-ADDR>/<MASK>

no ip address <IPV4-ADDR>/<MASK>
```

Description

Sets the local IP address of a GRE tunnel. This address identifies the tunnel interface for routing. It must be on the same subnet as the tunnel address assigned on the remote device.

The `no` form of this command deletes the local IP address assigned to a GRE tunnel.

Command context

```
config-gre-if
```

Parameters

<IPV4-ADDR>

Specifies the tunnel IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255. You can remove leading zeros. For example, the address `192.169.005.100` becomes `192.168.5.100`.

<MASK>

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 32.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defines the local IP address **`10.10.10.1`** for GRE tunnel **`33`**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# ip address 10.10.10.1/24
```

Deletes the local IP address **`10.10.10.1`** for GRE tunnel 33.

```
switch(config)# interface tunnel 33
switch(config-gre-if)# no ip address 10.10.10.1/24
```

ipv6 address

Syntax

```
ipv6 address <IPV6-ADDR>/<MASK>

no ipv6 address <IPV6-ADDR>/<MASK>
```

Description

Sets the local IP address of an IPv6 to IPv4 tunnel or of an IPv6 to IPv6 tunnel. This address identifies the tunnel interface for routing. It must be on the same subnet as the tunnel address assigned on the remote device.

The `no` form of this command deletes the local IP address assigned to an IPv6 to IPv4 tunnel.

Command context

`config-ip-if`

`config-if`

Parameters

<IPV6-ADDR>

Specifies the tunnel IP address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

<MASK>

Specifies the number of bits in the address mask in CIDR format (`x`), where `x` is a decimal number from 0 to 32.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defines the local IP address **`2001:DB8:5::1/64`** for tunnel **`8`** for an IPv6 to IPv6 tunnel.

```
switch(config)# interface tunnel 8 mode ip 6in6
switch(config-ip-if)# ipv6 address 2001:DB8:5::1/64
```

Deletes the local IP address **`2001:DB8::1/32`** for tunnel **`8`**.

```
switch(config)# interface tunnel 8
switch(config-ip-if)# no ipv6 address 2001:DB8:5::1/64
```

ip mtu

Syntax

`ip mtu <VALUE>`

Description

Sets the MTU (maximum transmission unit) for an IP interface. The default value is 1500 bytes.

The `no` form of this command sets the MTU to the default value of 1500 bytes.

Command context

`config-gre-if`

`config-ip-if`

Parameters

<VALUE>

Specifies the MTU in bytes. Range: 1,280 bytes to 9,192 bytes.

Authority

Administrators or local user group members with execution rights for this command.

Usage

The IP MTU is the largest IP packet that can be sent or received by the interface. For a tunnel, the IP MTU is the maximum size of the IP payload. To enable jumbo packet forwarding through the tunnel, set the IP MTU

of the tunnel to a value greater than 1500. Also set the MTU and the IP MTU values for the underlying physical interface that the tunnel is using to a value greater than 1,500 bytes. The IP MTU of the tunnel must also be greater than or equal to the MTU of the ingress interface on the switch. The IP MTU value of the tunnel must also be less than or equal to the IP MT of the underlying interface that the tunnel is using.

When defining a GRE tunnel, the MTU has to account for 28 bytes of IP layer overhead, plus a GRE header. It must be larger than the MTU of the interface that the tunnel is using. Packets larger than the MTU are dropped.

Examples

Sets the MTU on GRE interface `33` to **`1300`** bytes.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# mtu 1300
```

Sets the MTU on GRE interface `33` to the default value.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# ip mtu
```

Sets the MTU on IPv6 in IPv4 tunnel `27` to **`1000`** bytes.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)# mtu 1000
```

Sets the MTU onIPv6 in IPv4 tunnel `27` to the default value.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)# ip mtu
```

Sets the MTU on IPv6 in IPv6 tunnel `8` to **`900`** bytes.

```
switch(config)# interface tunnel 8 mode ip 6in6
switch(config-ip-if)# ip mtu 9000
```

Sets the MTU on IPv6 in IPv6 tunnel `8` to the default value.

```
switch(config)# interface tunnel 8 mode ip 6in6
switch(config-ip-if)# ip mtu
```

show interface tunnel

Syntax

`show interface tunnel[<TUNNEL-NUMBER>] [vsx-peer]`

Description

Shows configuration settings for all IP tunnels, or a specific tunnel.

Command context

Manager (#)

Parameters

<TUNNEL-NUMBER>

Specifies the number of an IP tunnel. Range: 1 to 127.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Shows configuration settings for tunnel **10**, which is a GRE tunnel in the following example.

```
switch# show interface tunnel10

Interface tunnel10 is up
 Admin state is up
 tunnel type GRE IPv4
 tunnel interface IPv4 address 192.0.2.0/24
 tunnel source IPv4 address 1.1.1.1
 tunnel destination IPv4 address 2.2.2.2
 tunnel ttl 60
RX
            0 input packets              0 bytes
            0 dropped

TX
            0 output packets             0 bytes
            0 dropped
```

Shows configuration settings for tunnel **12**, which is an IPv6 in IPv6 tunnel in the following example.

```
switch# show interface tunnel12

Interface tunnel12 is up
 Admin state is up
 tunnel type IPv6 in IPv6
 tunnel interface IPv6 address 4::1/64
 tunnel source IPv6 address 2::1
 tunnel destination IPv6 address 2::2
 tunnel ttl 60
 Description: Network2 Tunnel
RX
            0 input packets              0 bytes
            0 dropped

TX
            0 output packets             0 bytes
            0 dropped
```

Shows configuration settings for all tunnels.

```
switch# show interface tunnel

Interface tunnel10 is up
 Admin state is up
 tunnel type GRE IPv4
 tunnel interface IPv4 address 192.0.2.0/24
 tunnel source IPv4 address 1.1.1.1
 tunnel destination IPv4 address 2.2.2.2
 tunnel ttl 60
```

```
RX
            0 input packets               0 bytes
            0 dropped

TX
            0 output packets              0 bytes
            0 dropped

Interface tunnel11 is up
 Admin state is up
 tunnel type IPv6 in IPv4
 tunnel source IPv4 address 198.51.100.0
 tunnel destination IPv4 address 198.51.200.5
 tunnel ttl 80
 Description: Network11
RX
            0 input packets               0 bytes
            0 dropped

TX
            0 output packets              0 bytes
            0 dropped

Interface tunnel12 is up
 Admin state is up
 tunnel type IPv6 in IPv6
 tunnel interface IPv6 address 4::1/64
 tunnel source IPv6 address 2::1
 tunnel destination IPv6 address 2::2
 tunnel ttl 60
 Description: Network2 Tunnel
RX
            0 input packets               0 bytes
            0 dropped

TX
            0 output packets              0 bytes
            0 dropped
```

show running-config interface tunnel

Syntax

`show running-config interface tunnel<TUNNEL-NUMBER> [vsx-peer]`

Description

Shows the commands used to configure an IP tunnel.

Command context

`Manager (#)`

Parameters

<TUNNEL-NUMBER>

Specifies the number of an IP tunnel. Range: 1 to 127.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Shows the configuration for a GRE tunnel.

```
switch# show running-config interface tunnel2
  interface tunnel 2 mode gre ipv4
  source ip 10.10.20.11
  destination ip 10.20.1.2
  ip address 10.10.10.1/24
  ttl 60
```

Shows the configuration for IPv6 in IPv4 tunnel.

```
switch# show running-config interface tunnel5
  interface tunnel5 mode ip 6in4
  source ip 10.10.10.12
  destination ip 22.20.20.20
  ip6 address 2001:DB8:5::1/64
  ttl 60
  no shutdown
  description Network10
```

Shows the configuration for IPv6 in IPv6 tunnel.

```
switch# show running-config interface tunnel1
  interface tunnel 1 mode ip 6in6
  description Network2 Tunnel
  source ipv6 2::1
  destination ipv6 2::2
  ipv6 address 4::1/64
  ttl 60
```

shutdown

Syntax

```
shutdown
no shutdown
```

Description

This command disables an IP interface. IP interfaces are disabled by default when created.

The `no` form of this command enables an IP interface.

Command context

`config-gre-if`

`config-ip-if`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enables GRE interface **33**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# no shutdown
```

Disables GRE interface **33**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# shutdown
```

Enables IPv6 in IPv4 interface **27**.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)# no shutdown
```

Disables IPv6 in IPv4 interface **27**.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)# shutdown
```

source ip

Syntax

```
source ip <IPV4-ADDR>

no source ip <IPV4-ADDR>
```

Description

Sets the source IP address for an IP tunnel. Specify the IP address of a layer 3 interface on the switch. Tunnels can have the same source IP address and different destination IP addresses.

The `no` form of this command deletes the source IP address for an IP tunnel.

Command context

`config-gre-if`

`config-ip-if`

Parameters

<IPV4-ADDR>

Specifies the source IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defines the source IP address to be **10.10.20.1** for GRE tunnel **33**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# source ip 10.10.20.1
```

Deletes the source IP address **10.1.20.1** from GRE tunnel **33**.

```
switch(config)# interface tunnel 33
 switch(config-gre-if)# no source ip 10.10.20.1
```

Defines the source IP address to be **10.10.10.1** for IPv6 in IPv4 tunnel **27**.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)# source ip 10.10.10.1
```

Deletes the source IP address **10.1.10.1** from IPv6 in IPv4 tunnel **27**.

```
switch(config)# interface tunnel 27
 switch(config-ip-if)# no source ip 10.10.10.1
```

source ipv6

Syntax

```
source ipv6 <IPV6-ADDR>

no source ipv6
```

Description

Sets the source IPv6 address to be used for the encapsulation.

The `no` form of this command deletes the source IPv6 address for an IP tunnel.

Command context

```
config-ip-if
```

Parameters

<IPV6-ADDR>

Specifies the tunnel IP address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defines the source IPv6 address to be **2001:DB8::1** for IPv6 in IPv6 tunnel **8**.

```
switch(config)# interface tunnel 8 mode ip 6in6
switch(config-ip-if)# source ipv6 2001:DB8::1
```

Deletes the source IP address **2001:DB8::1** from IPv6 in IPv6 tunnel **8**.

```
switch(config)# interface tunnel 8
switch(config-ip-if)# no source ipv6 2001:DB8::1
```

ttl

Syntax

```
ttl <COUNT>

no ttl
```

Description

Sets the TTL (time-to-live), also known as the hop count, for tunneled packets. If not configured, the default value of 64 is used for the tunnel. (The hop count of the original packets is not changed.) A maximum of four different TTL values can be used at the same time by all tunnels on the switch. For example, if tunnel-1 has

TTL 10, tunnel-2 has TTL 20, tunnel-3 has TTL 30, and tunnel-4 has TTL 40, then tunnel-5 cannot have a unique TTL value, it must reuse one of the values assigned to the other tunnels (10, 20, 30, 40).

The `no` form of this command sets TTL to the default value of 64.

Command context

`config-gre-if`

`config-ip-if`

Parameters

<COUNT>

Specifies the hop count. Range: 1 to 255. Default: 64.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Defines a TTL of **99** for GRE tunnel **33**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# ttl 99
```

Sets the TTL for GRE tunnel **33** to the default value of 64.

```
switch(config)# interface tunnel 33
switch(config-gre-if)# no ttl
```

Defines a TTL of **55** for IPv6 in IPv4 tunnel **27**.

```
switch(config)# interface tunnel 27 mode ip 6in4
switch(config-ip-if)# ttl 55
```

Sets the TTL for IPv6 in IPv4 tunnel **27** to the default value of 64.

```
switch(config)# interface tunnel 27
switch(config-ip-if)# no ttl
```

vrf attach

Syntax

`vrf attach <VRF-NAME>`

`no vrf attach <VRF-NAME>`

Description

Assigns an IP tunnel to a VRF. By default, all tunnels are automatically assigned to the default VRF when they are created.

The `no` form of this command assigns a tunnel to the default VRF (`default`).

Command context

`config-gre-if`

`config-ip-if`

Parameters

<VRF-NAME>

Specifies the VRF name to which to assign the tunnel.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Assigns GRE tunnel **33** to **vrf1**.

```
switch(config)# interface tunnel 33 mode gre ipv4
switch(config-gre-if)# vrf attach vrf1
```

Reassigns GRE tunnel **33** to the default VRF.

```
switch(config)# interface tunnel 33
switch(config-gre-if)# no vrf attach vrf1
```

Assigns IPv6 in IPv4 tunnel **27** to **vrf2**.

```
switch(config)# interface tunnel 27 mode gre ipv4
switch(config-ip-if)# vrf attach vrf2
```

Reassigns IPv6 in IPv4 tunnel **27** to the default VRF.

```
switch(config)# interface tunnel 27
switch(config-ip-if)# no vrf attach vrf2
```

Assigns IPv6 in IPv6 tunnel **8** to **vrf3**.

```
switch(config)# interface tunnel 8 mode ip 6in6
switch(config-ip-if)# vrf attach vrf3
```

Reassigns IPv6 in IPv6 tunnel **8** to the default VRF.

```
switch(config)# interface tunnel 8
switch(config-ip-if)# no vrf attach vrf3
```

The Internet Control Message Protocol (ICMP) is a supporting protocol in the Internet protocol suite. The protocol is used by network devices, including routers, to send error messages and operational information. For example, an ICMP message might indicate that a requested service is not available. Another example of an ICMP message might be that a host or router could not be reached.

ICMP message types

The type field identifies the type of message sent by the host or gateway.

Type	ICMP messages
0	Echo Reply (Ping Reply, used with Type 8, Ping Request)
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request (Ping Request, used with Type 0, Ping Reply)
9	Router Advertisement (Used with Type 9)
10	Router Solicitation (Used with Type 10)
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request (Used with Type 14)
14	Timestamp Reply (Used with Type 13)
15	Information Request (obsolete) (Used with Type 16)
16	Information Reply (obsolete) (Used with Type 15)
17	Address Mask Request (Used with Type 17)
18	Address Mask Reply (Used with Type 18)

When ICMP messages are sent

ICMP messages are sent when one or more of the following scenarios occur:

- A datagram cannot reach its destination.
- The gateway does not have the buffering capacity to forward a datagram.
- The gateway can direct the host to send traffic on a shorter route.

ICMP redirect messages

ICMP redirect messages are used by routers to notify the hosts on the data link that a better route is available for a particular destination.

When ICMP redirect messages are sent

The switch is configured to send redirects by default. ICMP redirect messages are sent when one or more of the following scenarios occur:

- The interface on which the packet comes into the router is the same interface on which the packet gets routed out.
- The subnet or network of the source IP address is on the same subnet or network of the next-hop IP address of the routed packet.
- The datagram is not source-routed.
- The destination unicast address is unreachable. In this case, the router generates the ICMP destination unreachable message to inform the source host about the situation.

ICMP commands

ip icmp redirect

Syntax

```
ip icmp redirect
```

```
no ip icmp redirect
```

Description

Enables the sending of ICMPv4 and ICMPv6 redirect messages to the source host. Enabled by default.

The `no` form of this command disables ICMPv4 and ICMPv6 redirect messages to the source host.

Command context

```
config
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling ICMP redirect messages:

```
switch(config)# ip icmp redirect
```

Disabling ICMP redirect messages:

```
switch(config)# no ip icmp redirect
```

ip icmp throttle

Syntax

```
ip icmp throttle <packet-interval>
```

```
no ip icmp throttle
```

Description

Used to configure the throttle parameter for both ICMPv4 and ICMPv6 error messages and redirect messages.

The `no` form of this command disables the throttle parameter for both ICMPv4 and ICMPv6 error messages and redirect messages.

Command context

```
config
```

Parameters

<packet-interval>

Specifies the ICMPv4/v6 packet interval in seconds. Default: 1 second. Range: 1-86400.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling the throttle parameter for both ICMPv4 and ICMPv6 error messages and redirect messages:

```
switch(config)# ip icmp throttle 3000
```

Disabling the throttle parameter for both ICMPv4 and ICMPv6 error messages and redirect messages:

```
switch(config)# no ip icmp throttle
```

ip icmp unreachable

Syntax

```
ip icmp unreachable
```

```
no ip icmp unreachable
```

Description

Enables the sending of ICMPv4 and ICMPv6 destination unreachable messages on the switch to a source host when a specific host is unreachable. The unreachable host address originates from the failed packed. Default setting.

The `no` form of this command disables the sending of ICMPv4 and ICMPv6 destination unreachable messages from the switch to a source host when a specific host is unreachable. This command does not prevent other hosts from sending an ICMP unreachable message.

Command context

`config`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling ICMPv4 and ICMPv6 destination unreachable messages to a source host:

```
switch(config)# ip icmp unreachable
```

Disabling ICMPv4 and ICMPv6 destination unreachable messages to a source host:

```
switch(config)# no ip icmp unreachable
```

The Domain Name System (DNS) is the Internet protocol for mapping a hostname to its IP address. DNS allows users to enter more readily memorable and intuitive hostnames, rather than IP addresses, to identify devices connected to a network. It also allows a host to keep the same hostname even if it changes its IP address.

Hostname resolution can be either static or dynamic.

- In static resolution, a local table is defined on the switch that associates hostnames with their IP addresses. Static tables can be used to speed up the resolution of frequently queried hosts.
- Dynamic resolution requires that the switch query a DNS server located elsewhere on the network. Dynamic name resolution takes more time than static name resolution, but requires far less configuration and management.

DNS client

The DNS client resolves hostnames to IP addresses for protocols that are running on the switch. When the DNS client receives a request to resolve a hostname, it can do so in one of two ways:

- Forward the request to a DNS name server for resolution.
- Reply to the request without using a DNS name server, by resolving the name using a statically defined table of hostnames and their associated IP addresses.

Configuring the DNS client

Procedure

1. Configure one or more DNS name servers with the command `ip dns server`.
2. To resolve DNS requests by appending a domain name to the requests, either configure a single domain name with the command `ip dns domain-name`, or configure a list of up to six domain names with the command `ip dns domain-list`.
3. To use static name resolution for certain hosts, associate an IP address to a host with the command `ip dns host`.
4. Review your DNS configuration settings with the command `show ip dns`.

Examples

This example creates the following configuration:

- Defines the domain **`switch.com`** to append to all requests.
- Defines a DNS server with IPv4 address of **`1.1.1.1`**.

- Defines a static DNS host named **myhost1** with an IPv4 address of **3.3.3.3**.
- DNS client traffic is sent on the default VRF (named **default**).

```
switch(config)# ip dns domain-name switch.com
switch(config)# ip dns server-address 1.1.1.1
switch(config)# ip dns host myhost1 3.3.3.3
switch(config)# exit
switch# show ip dns

VRF Name : vrf_mgmt

Host Name                                                                       Address
-------------------------------------------------------------------------------------


VRF Name : vrf_default
Domain Name : switch.com
DNS Domain list :
Name Server(s) : 1.1.1.1

Host Name                                                                       Address
-------------------------------------------------------------------------------------
myhost1
```

This example creates the following configuration:

- Defines three domains to append to DNS requests **domain1.com**, **domain2.com**, **domain3.com** with traffic forwarding on VRF **mainvrf**.
- Defines a DNS server with an IPv6 address of **c::13**.
- Defines a DNS host named **myhost** with an IPv4 address of **3.3.3.3**.

```
switch(config)# ip dns domain-list domain1.com vrf mainvrf
switch(config)# ip dns domain-list domain2.com vrf mainvrf
switch(config)# ip dns domain-list domain3.com vrf mainvrf
switch(config)# ip dns server-address c::13
switch(config)# ip dns host myhost 3.3.3.3 vrf mainvrf
switch(config)# quit
switch# show ip dns mainvrf

VRF Name : mainvrf
Domain Name :
DNS Domain list : domain1.com, domain2.com, domain3.com
Name Server(s) : c::13

Host Name                                                                       Address
-------------------------------------------------------------------------------------
myhost                                                                          3.3.3.3
```

DNS client commands

ip dns domain-list

Syntax

```
ip dns domain-list <DOMAIN-NAME> [vrf <VRF-NAME>]

no ip dns domain-list <DOMAIN-NAME> [vrf <VRF-NAME>]
```

Description

Configures one or more domain names that are appended to the DNS request. The DNS client appends each name in succession until the DNS server replies. Domains can be either IPv4 or IPv6. By default, requests are forwarded on the default VRF.

The `no` form of this command removes a domain from the list.

Command context

```
config
```

Parameters

`list <DOMAIN-NAME>`

Specifies a domain name. Up to six domains can be added to the list. Length: 1 to 256 characters.

`vrf <VRF-NAME>`

Specifies a VRF name. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

This example defines a list with two entries: **`domain1.com`** and **`domain2.com`**.

```
switch(config)# ip dns domain-list domain1.com
switch(config)# ip dns domain-list domain2.com
```

This example defines a list with two entries, **`domain2.com`** and **`domain5.com`**, with requests being sent on **`mainvrf`**.

```
switch(config)# ip dns domain-list domain2.com vrf mainvrf
switch(config)# ip dns domain-list domain5.com vrf mainvrf
```

This example removes the entry **`domain1.com`**.

```
switch(config)# no ip dns domain-list domain1.com
```

ip dns domain-name

Syntax

```
ip dns domain-name <DOMAIN-NAME> [ vrf <VRF-NAME> ]

no ip dns domain-name <DOMAIN-NAME> [ vrf <VRF-NAME> ]
```

Description

Configures a domain name that is appended to the DNS request. The domain can be either IPv4 or IPv6. By default, requests are forwarded on the default VRF. If a domain list is defined with the command **ip dns domain-list**, the domain name defined with this command is ignored.

The no form of this command removes the domain name.

Command context

config

Parameters

<DOMAIN-NAME>

Specifies the domain name to append to DNS requests. Length: 1 to 256 characters.

vrf *<VRF-NAME>*

Specifies a VRF name. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Setting the default domain name to domain.com:

```
switch(config)# ip dns domain-name domain.com
```

Removing the default domain name domain.com:

```
switch(config)# no ip dns domain-name domain.com
```

ip dns host

Syntax

```
ip dns host <HOST-NAME> <IP-ADDR> [ vrf <VRF-NAME> ]

no ip dns host <HOST-NAME> <IP-ADDR> [ vrf <VRF-NAME> ]
```

Description

Associates a static IP address with a hostname. The DNS client returns this IP address instead of querying a DNS server for an IP address for the hostname. Up to six hosts can be defined. If no VRF is defined, the default VRF is used.

The no form of this command removes a static IP address associated with a hostname.

Command context

config

Parameters

host *<HOST-NAME>*

Specifies the name of a host. Length: 1 to 256 characters.

<IP-ADDR>

Specifies an IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

`vrf <VRF-NAME>`

Specifies a VRF name. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

This example defines an IPv4 address of **`3.3.3.3`** for **`host1`**.

```
switch(config)# ip dns host host1 3.3.3.3
```

This example defines an IPv6 address of **`b::5`** for **`host 1`**.

```
switch(config)# ip dns host host1 b::5
```

This example defines removes the entry for **`host 1`** with address **`b::5`**.

```
switch(config)# no ip dns host host1 b::5
```

ip dns server address

Syntax

```
ip dns server-address <IP-ADDR> [ vrf <VRF-NAME> ]

no ip dns server-address <IP-ADDR> [ vrf <VRF-NAME> ]
```

Description

Configures the DNS name servers that the DNS client queries to resolve DNS queries. Up to six name servers can be defined. The DNS client queries the servers in the order that they are defined. If no VRF is defined, the default VRF is used.

The `no` form of this command removes a name server from the list.

Command context

`config`

Parameters

<IP-ADDR>

Specifies an IP address in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255, or IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F.

`vrf <VRF-NAME>`

Specifies a VRF name. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

This example defines a name server at **1.1.1.1**.

```
switch(config)# ip dns server-address 1.1.1.1
```

This example defines a name server at **a::1**.

```
switch(config)# ip dns server-address a::1
```

This example removes a name server at **a::1**.

```
switch(config)# no ip dns server-address a::1
```

show ip dns

Syntax

```
show ip dns [vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows all DNS client configuration settings or the settings for a specific VRF.

Command context

Manager (#)

Parameters

vrf *<VRF-NAME>*

Specifies the VRF for which to show information. If no VRF is defined, the default VRF is used.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

These examples define DNS settings and then show how they are displayed with the `show ip dns` command.

```
switch(config)# ip dns domain-name domain.com
switch(config)# ip dns domain-list domain5.com
switch(config)# ip dns domain-list domain8.com
switch(config)# ip dns server-address 4.4.4.4
switch(config)# ip dns server-address 6.6.6.6
switch(config)# ip dns host host3 5.5.5.5
switch(config)# ip dns host host2 2.2.2.2
switch(config)# ip dns host host3 c::12
switch(config)# ip dns domain-name reddomain.com vrf red
switch(config)# ip dns domain-list reddomain5.com vrf red
switch(config)# ip dns domain-list reddomain8.com vrf red
switch(config)# ip dns server-address 4.4.4.5 vrf red
switch(config)# ip dns server-address 6.6.6.7 vrf red
switch(config)# ip dns host host3 5.5.5.6 vrf red
```

```
switch(config)# ip dns host host2 2.2.2.3 vrf red
switch(config)# ip dns host host3 c::13 vrf red
switch# show ip dns
VRF Name : default

Domain Name : domain.com
DNS Domain list : domain5.com, domain8.com
Name Server(s) : 4.4.4.4, 6.6.6.6

Host Name    Address
------------------------------
host2            2.2.2.2
host3            5.5.5.5
host3            c::12

VRF Name : red

Domain Name : reddomain.com
DNS Domain list : reddomain5.com, reddomain8.com
Name Server(s) : 4.4.4.5, 6.6.6.7

Host Name    Address
------------------------------
host2            2.2.2.3
host3            5.5.5.6
host3            c::13
```

```
switch(config)# ip dns domain-name domain.com vrf red
switch(config)# ip dns domain-list domain5.com vrf red
switch(config)# ip dns domain-list domain8.com vrf red
switch(config)# ip dns server-address 4.4.4.4 vrf red
switch(config)# ip dns server-address 6.6.6.6 vrf red
switch(config)# ip dns host host3 5.5.5.5 vrf red
switch(config)# no ip dns host host2 2.2.2.2 vrf red
switch(config)# ip dns host host3 c::12 vrf red

switch# show ip dns vrf red
VRF Name : red

Domain Name : domain.com
DNS Domain list : domain5.com, domain8.com
Name Server(s) : 4.4.4.4, 6.6.6.6

Host Name    Address
------------------------------
host3            5.5.5.5
host3            c::12
```

ARP (Address Resolution Protocol) is used to map the network address assigned to a device to its physical address. For example, on an Ethernet network, ARP maps layer 3 IPv4 network addresses to layer 2 MAC addresses. (ARP does not work with IPv6 addresses. Instead, the Neighbor discovery protocol is used.)

ARP operates at layer 2. ARP requests are broadcast to all devices on the local network segment and are not forwarded by routers. ARP is enabled by default and cannot be disabled.

Proxy ARP

Proxy ARP allows a routing switch to answer ARP requests from devices on one network on behalf of devices on another network. The ARP proxy is aware of the location of the traffic destination, and offers its own MAC address as the final destination.

For example, if Proxy ARP is enabled on a routing switch connected to two subnets (10.10.10.0/24 and 20.20.20.0/24), the routing switch can respond to an ARP request from 10.10.10.69 for the MAC address of the device with IP address 20.20.20.69.

Typically, the host that sent the ARP request then sends its packets to the switch that has the ARP proxy. This switch then forwards the packets to the intended host through a mechanism such as a tunnel.

Proxy ARP is supported on L3 physical and VLAN interfaces. It is disabled by default. To enable proxy ARP, routing must be enabled on the interface.

Local proxy ARP

Local proxy ARP is a technique by which a device on a given network answers the ARP queries for a host address that is on the same network. It is primarily used to enable layer 3 communication between hosts within a common subnet that are separated by layer 2 boundaries (Example: PVLAN). Local proxy ARP is supported on L3 physical and VLAN interfaces.

Local proxy ARP is disabled by default. Routing must be enabled on the interface to enable local proxy ARP.

Dynamic ARP Inspection

ARP is used for resolving IP against MAC addresses on a broadcast network segment like the Ethernet and was originally defined by Internet Standard RFC 826. ARP does not support any inherent security mechanism and as such depends on simple datagram exchanges for the resolution, with many of these being broadcast.

Because it is an unreliable and non-secure protocol, ARP is vulnerable to attacks. Some attacks may be targeted toward the networks whereas other attacks may be targeted toward the switch itself. The attacks primarily intend to create denial of service (DoS) for the other entities present in the network.

Most of the attacks are carried out in one of the following three forms:

- Overwhelming the switch control plane with too many ARP packets.
- Overwhelming the switch control plane with too many unresolved data packets.
- Masquerading as a trusted gateway/server by wrongly advertising ARPs.

Several defense mechanisms can be put in place on a switch to protect against attacks:

- Limit the amount of ARP activity allowed from a host or on a port.
- Ensure that all ARP packets are consistent with one or more binding databases, which can be created through various means.
- Enforce integrity checks on the ARP packets to check against different MAC or IP addresses in the Ethernet or IP header and ARP header.

This release implements Dynamic ARP Inspection to enforce DHCP snooping binding on all ARP packets and is limited to the 8400 platform. The feature will be disabled from the code, CLI, and schema by the use of appropriate config flags for other platforms.

Only the following is supported:

- Enabling and disabling of Dynamic ARP Inspection on a VLAN level (it does not have to be SVI).
- Defining the member ports of a VLAN as either trusted or untrusted.
- Only ARP traffic on untrusted ports subjected to checks.
- Routed ports (RoPs) always treated as trusted.
- Listening to the DHCP Bindings table and check every ARP packet to match against the binding.

ARP ACLs are not supported in this release and the DHCP snooping table will be the only source of binding.

Configuring proxy ARP

Procedure

1. Switch to configuration context with the command `config`.
2. Switch to an interface with the command `interface`, or to an interface VLAN with the command `interface vlan`, or to a LAG with the command `interface lag`.
3. Enable local proxy ARP with the command `ip proxy-arp`.

Examples

This example configures proxy ARP on interface **1/1/2**

```
switch# config
switch(config)# interface 1/1/2
switch(config-if)# ip proxy-arp
```

.

This example configures proxy ARP on interface VLAN **30**.

```
switch# config
switch(config)# interface vlan 30
switch(config-vlan-30)# ip proxy-arp
```

Configuring local proxy ARP

Procedure

1. Switch to configuration context with the command `config`.
2. Switch to an interface with the command `interface`, or to an interface VLAN with the command `interface vlan`, or to a LAG with the command `interface lag`.
3. Enable local proxy ARP with the command `ip local-proxy-arp`.

Examples

This example configures local proxy ARP on interface **1/1/2**

```
switch# config
switch(config)# interface 1/1/2
switch(config-if)# ip local-proxy-arp
```

.

This example configures local proxy ARP on interface VLAN **30**.

```
switch# config
switch(config)# interface vlan 30
switch(config-vlan-30)# ip local-proxy-arp
```

Dynamic ARP Inspection

ARP is used for resolving IP against MAC addresses on a broadcast network segment like the Ethernet and was originally defined by Internet Standard RFC 826. ARP does not support any inherent security mechanism and as such depends on simple datagram exchanges for the resolution, with many of these being broadcast.

Because it is an unreliable and non-secure protocol, ARP is vulnerable to attacks. Some attacks may be targeted toward the networks whereas other attacks may be targeted toward the switch itself. The attacks primarily intend to create denial of service (DoS) for the other entities present in the network.

Most of the attacks are carried out in one of the following three forms:

- Overwhelming the switch control plane with too many ARP packets.
- Overwhelming the switch control plane with too many unresolved data packets.
- Masquerading as a trusted gateway/server by wrongly advertising ARPs.

Several defense mechanisms can be put in place on a switch to protect against attacks:

- Limit the amount of ARP activity allowed from a host or on a port.
- Ensure that all ARP packets are consistent with one or more binding databases, which can be created through various means.
- Enforce integrity checks on the ARP packets to check against different MAC or IP addresses in the Ethernet or IP header and ARP header.

This release implements Dynamic ARP Inspection to enforce DHCP snooping binding on all ARP packets and is supported on the 6300, 6400, and 8400 platforms. The feature will be disabled from the code, CLI, and schema by the use of appropriate config flags for other platforms.

Only the following is supported:

- Enabling and disabling of Dynamic ARP Inspection on a VLAN level (it does not have to be SVI).
- Defining the member ports of a VLAN as either trusted or untrusted.
- Only ARP traffic on untrusted ports subjected to checks.
- Routed ports (RoPs) always treated as trusted.
- Listening to the DHCP Bindings table and check every ARP packet to match against the binding.

ARP ACLs are not supported in this release and the DHCP snooping table will be the only source of binding.

ARP commands

arp cache-limit

Syntax

`arp cache-limit <LIMIT>`

Description

Specifies the maximum number of entries in the ARP (Address Resolution Protocol) cache.

Command context

`config`

Parameters

<LIMIT>

Specifies the maximum number of entries in the ARP cache. Range: 4096 to 131072. Default: 131072.

Authority

Administrators or local user group members with execution rights for this command.

Examples

```
switch(config)# arp cache-limit 4097
```

arp inspection

Syntax

`arp inspection`

Description

Enables Dynamic ARP Inspection on the current VLAN, forcing all ARP packets from untrusted ports to be subjected to a MAC-IP association check against a binding table.

The `no` form of this command disables Dynamic ARP Inspection on the VLAN.

Command context

`config-vlan-<VLAN-ID>`

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling dynamic ARP inspection:

```
switch# configure terminal
switch(config)# vlan 1
switch(config-vlan)# arp inspection
```

Disabling dynamic ARP inspection:

```
switch# configure terminal
switch(config)# vlan 1
switch(config-vlan)# no arp inspection
```

arp inspection trust

Syntax

```
arp inspection trust

no arp inspection trust
```

Description

Configures the interface as a trusted. All interfaces are untrusted by default.

The `no` form of this command returns the interface to the default state (untrusted).

Command context

`config-if`

Authority

Administrators or local user group members with execution rights for this command.

Example

Setting an interface as trusted:

```
switch(config-if)# arp inspection trust
```

arp ipv4 mac

Syntax

```
arp ipv4 <IPV4_ADDR> mac <MAC_ADDR>

no arp ipv4 <IPV4_ADDR> mac <MAC_ADDR>
```

Description

Specifies a permanent static neighbor entry in the ARP table (for IPv4 neighbors).

The `no` form of this command deletes a permanent static neighbor entry from the ARP table.

Command context

`config-if`

`config-if-vlan`

Parameters

`ipv4 <IPV4-ADDR>`

Specifies the IP address of the neighbor or the virtual IP address of the cluster in IPv4 format (`x.x.x.x`), where `x` is a decimal number from 0 to 255. . Range: 4096 to 131072. Default: 131072.

`mac <MAC-ADDR>`

Specifies the MAC address of the neighbor or the multicast MAC address in IANA format (`xx:xx:xx:xx:xx:xx`), where `x` is a hexadecimal number from 0 to F. Range: 4096 to 131072. Default: 131072.

Authority

Administrators or local user group members with execution rights for this command.

Example

Configuring a static ARP entry on a interface VLAN **`10`**:

```
switch(config)# interface vlan 10
switch(config-if-vlan)# arp ipv4 2.2.2.2 mac 01:00:5e:00:00:01
```

Removing a static ARP entry on interface VLAN**`10`**:

```
switch(config)# interface vlan 10
switch(config-if-vlan)# no arp ipv4 2.2.2.2 mac 01:00:5e:00:00:01
```

clear arp

Syntax

`clear arp [port <PORT-ID> | vrf {all-vrfs | <VRF-NAME>}]`

Description

Clears IPv4 and IPv6 neighbor entries from the ARP table. If you do not specify any parameters, ARP table entries are cleared for the default VRF.

Command context

Manager (`#`)

Parameters

`port <PORT-ID>`

Specifies a physical port on the switch. Format: `member/slot/port`. For example: `1/1/1`. .

`all-vrfs`

Selects all VRFs.

`<VRF-NAME>`

Specifies the name of a VRF. Default: default.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Clearing all IPv4 and IPv6 neighbor ARP entries for the default VRF:

```
switch# clear arp
```

Clearing all ARP neighbor entries for a port ():

```
switch# clear arp 1/1/35
```

Clearing all IPv4 and IPv6 neighbor ARP entries for all VRFs:

```
switch# clear arp vrf all-vrfs
```

Clearing all IPv4 and IPv6 neighbor ARP entries for a specific VRF instance:

```
switch# clear arp vrf RED
```

ip local-proxy-arp

Syntax

```
ip local-proxy-arp

no ip local-proxy-arp
```

Description

Enables local proxy ARP on the specified interface. Local proxy ARP is supported on Layer 3 physical interfaces and on VLAN interfaces. To enable local proxy ARP on an interface, routing must be enabled on that interface.

The `no` form of this command disables local proxy ARP on the specified interface.

Command context

```
config-if

config-if-vlan
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling local proxy ARP on interface **1/1/1**:

```
switch# interface 1/1/1
switch(config-if)# ip local proxy-arp
```

Enabling local proxy ARP on interface VLAN **3**:

```
switch# interface vlan 3
switch(config-if-vlan)# ip local-proxy-arp
```

Disabling local proxy ARP on on interface **1/1/1**.

```
switch# interface 1/1/1
switch(config-if)# no ip local-proxy-arp
```

ipv6 neighbor mac

Syntax

```
ipv6 neighbor <IPV6-ADDR> mac <MAC-ADDR>

no ipv6 neighbor <IPV6-ADDR> mac <MAC-ADDR>
```

Description

Specifies a permanent static neighbor entry in the ARP table (for IPv6 neighbors).

The `no` form of this command deletes a permanent static neighbor entry from the ARP table.

Command context

```
config-if
```

Parameters

<IPV6-ADDR>>

Specifies an IP address in IPv6 format (`xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`), where `x` is a hexadecimal number from 0 to F. Range: 4096 to 131072. Default: 131072.

mac *<MAC-ADDR>>*

Specifies the MAC address of the neighbor (`xx:xx:xx:xx:xx:xx`), where `x` is a hexadecimal number from 0 to F. Range: 4096 to 131072. Default: 131072.

Authority

Administrators or local user group members with execution rights for this command.

Example

Creates a static ARP entry on interface `1/1/1`.

```
switch(config)# interface 1/1/1
switch(config-if)# arp ipv6 neighbor 2001:0db8:85a3::8a2e:0370:7334 mac 00:50:56:96:df:c8
```

ip proxy-arp

Syntax

```
ip proxy-arp

no ip proxy-arp
```

Description

Enables proxy ARP for the specified Layer 3 interface. Proxy ARP is supported on Layer 3 physical interfaces, LAG interfaces, and VLAN interfaces. It is disabled by default. To enable proxy ARP on an interface, routing must be enabled on that interface.

The `no` form of this command disables proxy ARP for the specified interface.

Command context

```
config-if

config-if-vlan

config-lag-vlan
```

Authority

Administrators or local user group members with execution rights for this command.

Examples

Enabling proxy ARP on interface **1/1/1**:

```
switch# interface 1/1/1
switch(config-if)# ip proxy-arp
```

Enabling proxy ARP on VLAN **3**:

```
switch# interface vlan 3
switch(config-if-vlan)# ip proxy-arp
```

Enabling proxy ARP on a LAG **11**:

```
switch(config)# int lag 11
switch(config-lag-if)# ip proxy-arp
```

Disabling proxy ARP on interface 1/1/1:

```
switch# interface 1/1/1
switch(config-if)# no ip proxy-arp
```

show arp

Syntax

```
show arp [vsx-peer]
```

Description

Shows the entries in the ARP (Address Resolution Protocol) table.

Command context

Manager (#)

Parameters

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Administrators or local user group members with execution rights for this command.

Usage

This command displays information about ARP entries, including the IP address, MAC address, port, and state.

When no parameters are specified, the `show arp` command shows all ARP entries for the default VRF (Virtual Router Forwarding) instance.

Examples

```
switch# show arp

IPv4 Address     MAC                Port         Physical
Port                        State
--------------------------------------------------------------------------------
192.168.1.2          00:50:56:96:7b:e0  vlan10        1/1/29           stale
192.168.1.3          00:50:56:96:7b:ac  vlan10        1/1/1            reachable

Total Number Of ARP Entries Listed- 2.
--------------------------------------------------------------------------------
```

show arp inspection interface

Syntax

```
show arp inspection interface
```

Description

Displays the current configuration of dynamic ARP inspection on a VLAN or interface.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show arp inspection interface

-------------------------------------------------------------------------------
Interface           Trust-State
-------------------------------------------------------------------------------
1/1/1               Untrusted
-------------------------------------------------------------------------------
```

```
switch# show arp inspection interface vsx-peer

-------------------------------------------------------------------------------
Interface           Trust-State
-------------------------------------------------------------------------------
1/1/1               Untrusted
lag100              Trusted
-------------------------------------------------------------------------------
```

```
switch# show arp inspection interface 1/1/1

-------------------------------------------------------------------------------
Interface           Trust-State
-------------------------------------------------------------------------------
1/1/1               Untrusted
-------------------------------------------------------------------------------
```

show arp inspection statistics

Syntax

```
show arp inspection statistics
```

Description

Displays statistics about forwarded and dropped ARP packets.

Command context

Operator (>) or Manager (#)

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show arp inspection statistics vlan 1-200

---------------------------------------------------------------------
VLAN    Name              Forwarded           Dropped
---------------------------------------------------------------------
1       DEFAULT_VLAN_1    0                   0
---------------------------------------------------------------------
```

```
switch# show arp inspection statistics vlan vsx-peer

---------------------------------------------------------------------
VLAN    Name              Forwarded           Dropped
---------------------------------------------------------------------
1       DEFAULT_VLAN_1    0                   0
200     VLAN200           0                   0
---------------------------------------------------------------------
```

show arp state

Syntax

```
show arp state {all | failed | incomplete | permanent | reachable | stale} [vsx-peer]
```

Description

Shows ARP (Address Resolution Protocol) cache entries that are in the specified state.

Command context

Operator (>) or Manager (#)

Parameters

`all`

Shows the ARP cache entries for all VRF (Virtual Router Forwarding) instances.

`failed`

Shows the ARP cache entries that are in `failed` state. The neighbor might have been deleted.

incomplete

Shows the ARP cache entries that are in `incomplete` state.

An incomplete state means that address resolution is in progress and the link-layer address of the neighbor has not yet been determined. A solicitation request was sent, and the switch is waiting for a solicitation reply or a timeout.

permanent

Shows the ARP cache entries that are in `permanent` state. ARP entries that are in a permanent state can be removed by administrative action only.

reachable

Shows the ARP cache entries that are in `reachable` state, meaning that the neighbor is known to have been reachable recently.

stale

Shows ARP cache entries that are in `stale` state.

ARP cache entries are in the `stale` state if the elapsed time is in excess of the ARP timeout in seconds since the last positive confirmation that the forwarding path was functioning properly.

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

```
switch# show arp state failed

IPv4 Address     MAC                Port         Physical Port    State
-------------------------------------------------------------------------
192.168.1.4                         vlan10                         failed
```

show arp summary

Syntax

```
show arp summary [all-vrfs | vrf <VRF-NAME>] [vsx-peer]
```

Description

Shows a summary of the IPv4 and IPv6 neighbor entries on the switch for all VRFs or a specific VRF.

Command context

Operator (>) or Manager (#)

Parameters

`all-vrfs`

Selects all VRFs.

`vrf <VRF-NAME>`

Specifies the name of a VRF.

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing summary ARP information for all VRFs:

```
switch# show arp summary all-vrfs

ARP Entry's State                : IPv4     IPv6
-----------------------------------------------------

Number of Reachable ARP entries  : 2         0

Number of Stale ARP entries      : 0         0

Number of Failed ARP entries     : 2         2

Number of Incomplete ARP entries : 0         0

Number of Permanent ARP entries  : 0         0

-----------------------------------------------------

Total ARP Entries: 6             : 4         2

-----------------------------------------------------
```

Showing a summary of all IPv4 and IPv6 neighbor entries on the primary and secondary (peer) switches:

```
vsx-primary# show arp summary
ARP Entry's State                  IPv4        IPv6
-----------------------------------------------------------
Number of Reachable ARP entries   25858       32231
Number of Stale ARP entries       0           1
Number of Failed ARP entries      0           257
Number of Incomplete ARP entries  0           0
Number of Permanent ARP entries   0           0
-----------------------------------------------------------
Total ARP Entries- 58347          25858       32489

vsx-primary# show arp summary vsx-peer
ARP Entry's State                  IPv4        IPv6
-----------------------------------------------------------
Number of Reachable ARP entries   25858       32168
Number of Stale ARP entries       0           3
Number of Failed ARP entries      0           317
Number of Incomplete ARP entries  0           0
```

```
Number of Permanent ARP entries   0           0
-----------------------------------------------------------
Total ARP Entries- 58346          25858       32488
-----------------------------------------------------------
```

show arp timeout

Syntax

```
show arp timeout [<INTERFACE>][vsx-peer]
```

Description

Shows the age-out period for each ARP (Address Resolution Protocol) entry for a port, LAG, or VLAN interface.

Command context

Operator (>) or Manager (#)

Parameters

<INTERFACE>

Specifies a physical port, VLAN, or LAG on the switch. For physical ports, use the format `member/slot/port` (for example, `1/3/1`).

[vsx-peer]

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing ARP timeout information for a VLAN:

```
switch# show arp timeout vlan4
```

Showing ARP timeout information for a port:

```
switch# show arp timeout 1/1/1
ARP Timeout:

------------------

Port              VRF                               Timeout

1/1/1             default                           600
```

show arp vrf

Syntax

```
show arp {all-vrfs | vrf <VRF-NAME>} [vsx-peer]
```

Description

Shows the ARP table for all VRF instances, or for the named VRF.

Command context

Operator (>) or Manager (#)

Parameters

`all-vrfs`

Specifies all VRFs.

`vrf <VRF-NAME>`

Specifies the name of a VRF. Length: 1 to 32 alphanumeric characters.

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Examples

Showing ARP entries for VRF **`test`**.

```
switch# show arp vrf test
ARP IPv4 Entries:
------------------------------------------------------
IPv4 Address    MAC                Port    Physical Port  State      VRF
10.20.30.40     00:50:56:bd:6a:c5  1/1/29  1/1/29         reachable  test
------------------------------------------------------
Total Number Of ARP Entries Listed: 1.
------------------------------------------------------

switch# show arp all-vrfs
ARP IPv4 Entries:
------------------------------------------------------
IPv4 Address    MAC                Port    Physical Port  State      VRF
192.168.120.10  00:50:56:bd:10:be  1/1/32  1/1/32         reachable  red
10.20.30.40     00:50:56:bd:6a:c5  1/1/29  1/1/29         reachable  test
------------------------------------------------------
Total Number Of ARP Entries Listed: 2.
------------------------------------------------------
```

Showing ARP entries for all VRFs.

```
switch# show arp all-vrfs
ARP IPv4 Entries:
------------------------------------------------------
IPv4 Address    MAC                Port    Physical Port  State      VRF
192.168.120.10  00:50:56:bd:10:be  1/1/32  1/1/32         reachable  red
10.20.30.40     00:50:56:bd:6a:c5  1/1/29  1/1/29         reachable  test
------------------------------------------------------
Total Number Of ARP Entries Listed: 2.
------------------------------------------------------
```

show ipv6 neighbors

Syntax

```
show ipv6 neighbors {all-vrfs | vrf <VRF-NAME>} [vsx-peer]
```

Description

Shows entries in the ARP table for all IPv6 neighbors for all VRFs or for a specific VRF.

When no parameters are specified, this command shows all ARP entries for the default VRF, and state information for `reachable` and `stale` entries only.

Command context

Operator (>) or Manager (#)

Authority

Administrators or local user group members with execution rights for this command.

Parameters

`all-vrfs`

Specifies all VRFs.

`vrf <VRF-NAME>`

Specifies a VRF name. Length: 1 to 32 alphanumeric characters.

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Examples

```
switch# show ipv6 neighbors
IPv6 Entries:

-------------------------------------------------------

IPv6 Address                  MAC                Port      Physical Port  State

fe80::a21d:48ff:fe8f:2700     a0:1d:48:8f:27:00  vlan2300  1/1/31         reachable

fe80::f603:43ff:fe80:a600     f4:03:43:80:a6:00  vlan2300  1/1/30         reachable

-------------------------------------------------------

Total Number Of IPv6 Neighbors Entries Listed: 2.

-------------------------------------------------------
```

show ipv6 neighbors state

Syntax

```
show ipv6 neighbors state {all | failed | incomplete | permanent | reachable | stale} [vsx-peer]
```

Description

Shows all IPv6 neighbor ARP (Address Resolution Protocol) cache entries, or those cache entries that are in the specified state.

Command context

Operator (>) or Manager (#)

Parameters

`all`

Shows all ARP cache entries.

`failed`

Shows ARP cache entries that are in `failed` state. The neighbor might have been deleted. Set the neighbor to be unreachable.

`incomplete`

Shows ARP cache entries that are in `incomplete` state.

An incomplete state means that address resolution is in progress and the link-layer address of the neighbor has not yet been determined. This means that a solicitation request was sent, and you are waiting for a solicitation reply or a timeout.

`permanent`

Shows ARP cache entries that are in `permanent` state.

`reachable`

Shows ARP cache entries that are in `reachable` state, meaning that the neighbor is known to have been reachable recently.

`stale`

Shows ARP cache entries that are in `stale` state.

ARP cache entries are in the `stale` state if the elapsed time is in excess of the ARP timeout in seconds since the last positive confirmation that the forwarding path was functioning properly.

`[vsx-peer]`

Shows the output from the VSX peer switch. If the switches do not have the VSX configuration or the ISL is down, the output from the VSX peer switch is not displayed. This parameter is available on switches that support VSX.

Authority

Operators or Administrators or local user group members with execution rights for this command. Operators can execute this command from the operator context (>) only.

Example

```
switch# show ipv6 neighbors state all

IPv6 Address                    MAC                Port     Physical Port    State
--------------------------------------------------------------------------------
100::2                          48:0f:cf:af:f1:cc  lag1     lag1             reachable
300::3                          48:0f:cf:af:33:be  vlan3    1/4/20           reachable
fe80::4a0f:cfff:feaf:f1cc       48:0f:cf:af:f1:cc  lag1     lag1             reachable
200::3                          48:0f:cf:af:33:be  1/4/11   1/4/11           reachable
fe80::4a0f:cfff:feaf:33be       48:0f:cf:af:33:be  vlan3    1/4/20           reachable
```

```
Total Number Of IPv6 Neighbors Entries Listed- 5.
--------------------------------------------------------------------------------
```

Overview

Network Load Balancing (NLB) is a load balancing technology for server clustering developed on Microsoft Windows Server. NLB supports load sharing and redundancy among servers within a cluster. To implement fast failover, NLB requires that the switch forwards network traffic to one or all servers in the cluster. Each server filters out the unexpected traffic. For more information, see **Configuring network infrastructure to support the NLB operation mode**

NLB is used to spread incoming requests across as many as 32 servers. Currently, the NLB in ArubaOS-CX switch supports only IGMP multicast mode. The IGMP multicast mode sends the packets out of the ports which connect to the cluster members. Assign a static multicast MAC address within the Internet Assigned Numbers Authority (IANA) range to the cluster's virtual unicast IP address. The clustered servers send IGMP joins to the configured multicast cluster group. If IGMP snooping is enabled, the switch dynamically populates the IGMP snooping table with the clustered servers, which prevents unicast flooding.

NLB commands

arp ipv4 mac

Syntax

```
arp ipv4 <IPv4-ADDR> mac <MAC-ADDR>

no arp ipv4 <IPv4-ADDR> mac <MAC-ADDR>
```

Description

Configures static ARP multicast on the interface.

The `no` form of this command removes the static ARP multicast configuration.

Command context

`config-if` and `config-if-vlan`

Parameters

<IPv4-ADDR>

Specifies cluster's virtual IPv4 address.

<MAC-ADDR>

Specifies multicast MAC address in IANA format (xx:xx:xx:xx:xx:xx) and non IANA format (xxxx.xxxx.xxxx).

Authority

Administrators or local user group members with execution rights for this command.

Examples

Configuring static ARP multicast on an interface:

```
switch(config)# vlan 10
switch(config-vlan-10)# no shutdown
switch(config-vlan-10)# ip igmp snooping enable
switch(config-vlan-10)# exit
switch(config)# interface vlan10
switch(config-if-vlan)# ip igmp enable
switch(config-if-vlan)# arp ipv4 10.1.30.254 mac 01:00:5e:7F:1E:FE
```

NOTE: If your NLB Virtual IP address is 10.1.30.254, then the server will join the 239.255.30.254 IGMP group. This IGMP group is mapped to the destination MAC address of 01:00:5e:7F:1E:FE.

show arp

Syntax

```
show arp
```

Description

Displays the static ARP multicast information.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Displaying the static ARP multicast information:

```
switch# show arp

IPv4 Address     MAC                Port         Physical Port    State
-------------------------------------------------------------------------
3.3.3.3          01:00:5e:00:00:02               1/1/1            permanent
2.2.2.2          01:00:5e:00:00:01  vlan10                        permanent

Total Number Of ARP Entries Listed- 2.
-------------------------------------------------------------------------
```

show ip igmp snooping vlan group

Syntax

```
show ip igmp snooping vlan <VLAN-ID> group IGMP-Group
```

Description

Displays multicast joins (members of the cluster) participating in the IGMP group.

Authority

Administrators or local user group members with execution rights for this command.

Examples

Displaying multicast joins participating in the IGMP group:

```
switch# show ip igmp snooping vlan 10 group 239.255.30.254

VLAN ID   : 10
VLAN Name : VLAN10
```

```
Group Address : 239.255.30.254
Last Reporter : 10.1.30.254
Group Type    : Filter

                                          V1        V2        Sources   Sources
Port      Vers Mode Uptime    Expires     Timer     Timer     Forwarded Blocked
--------- ---- ---- --------- --------- --------- --------- --------- --------
1/1/6     2    EXC  0m 21s    1m 12s              2m 48s    0         0
```

Accessing Aruba Support

Aruba Support Services	**https://www.arubanetworks.com/support-services/**
Aruba Support Portal	**https://asp.arubanetworks.com/**
North America telephone	1-800-943-4526 (US & Canada Toll-Free Number) +1-408-754-1200 (Primary - Toll Number) +1-650-385-6582 (Backup - Toll Number - Use only when all other numbers are not working)
International telephone	**https://www.arubanetworks.com/support-services/contact-support/**

Be sure to collect the following information before contacting Support:

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Other useful sites

Other websites that can be used to find information:

Airheads social forums and Knowledge Base	**https://community.arubanetworks.com/**
Software licensing	**https://lms.arubanetworks.com/**
End-of-Life information	**https://www.arubanetworks.com/support-services/end-of-life/**
Aruba software and documentation	**https://asp.arubanetworks.com/downloads**

Accessing updates

To download product updates:

Aruba Support Portal

https://asp.arubanetworks.com/downloads

If you are unable to find your product in the Aruba Support Portal, you may need to search My Networking, where older networking products can be found:

My Networking

https://www.hpe.com/networking/support

To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

https://support.hpe.com/portal/site/hpsc/aae/home/

IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.

To subscribe to eNewsletters and alerts:

https://asp.arubanetworks.com/notifications/subscriptions (requires an active Aruba Support Portal (ASP) account to manage subscriptions). Security notices are viewable without an ASP account.

Warranty information

To view warranty information for your product, go to **https://www.arubanetworks.com/support-services/product-warranties/**.

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at **https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts**

Additional regulatory information

Aruba is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements, environmental data (company programs, product recycling, energy efficiency), and safety information and compliance data, (RoHS and WEEE). For more information, see **https://www.arubanetworks.com/company/about-us/environmental-citizenship/**.

Documentation feedback

Aruba is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (**docsfeedback-switching@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.