CCIE Routing and Switching study notes

Christian Kyony

Version 1.0, 2015-10-12

Table of Contents

Dedication

To Cyril "Matiere" Kalenga

Part I : Layer 2 Technologies

Chapter 1. Switch administration

1.1. Interface Characteristics

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_2_se/configuration/guide/3750x_cg/swint.html

1.2. System clock

- Can be set manually or dynamic with NTP
- Keeps track internally based on UTC

Task: Configure the Time Zone

```
(config)# clock timezone <zone> <hours-offset> [minutes-offset]
```

Use the minutes-offset variable when the local time zone is a percentage of an hour different from UTC. For example, the time zone for some sections of Atlantic Canada (AST) is UTC-3.5, where the 3 means 3 hours and.5 means 50 percent. In this case, the necessary command is clock timezone AST -3 30.

Task: Reset the time to UTC

```
(config)# no clock timezone
```

Task: Set the system clock manually

```
clock set hh:mm:ss month day year
```

Task: Display the time and date configuration

```
# sh clock [detail]
```

Task: Configure Daylight Saving Time

```
(config)# clock summer-time <zone> recurring [week day month hh : mm : week day month
hh : mm [offset]]

Example(config)# clock summer-time PDT recurring 1 Sunday April 2:00 last Sunday
Octorber 2:00
```

1.3. System Name and Prompt

Task: Configure a system name

```
(config)# hostname <name>
```

1.4. MOTD login Banner

- MOTD and login not configured by default

Task: Configure a message of the day login banner

```
(config)# banner motd <delimiting-character>
<message> <delimiting-character>
```

1.5. Login Banner

- displayed on all connected terminals
- appears after the MOTD banner and before the login prompt

Task: Configure a Login Banner

```
(config)# banner login <delimiting-character>
<message> <delimiting-character>
```

1.6. MAC address table

- lists the destination MAC address with the associated VLANs , port number, and the type (static or dynamic)
- dynamic address are discarded after the **aging time** (300 seconds by default)

Task: Display Address Table Entries for the specified MAC address

```
# sh mac address-table address <MAC>
```

Task: Display only dynamic MAC addresses

```
# sh mac address-table dynamic
```

Task: Display the number of addresses present

```
# sh mac address-table count
```

Task: Display the MAC address table information for the specified VLAN

```
# sh mac address-table vlan
```

Task: Display the MAC address table information for the specified interface

```
# sh mac address-table interface
```

1.6.1. Aging time

- Default: 300 seconds

Task: Set the length of time that a dynamic entry remains in the MAC address after the entry is used or updated

```
# mac address-table aging-time [0 | 10-1000000] [vlan <1-4094>]
```

Task: Displays the aging time

```
# sh mac address-table aging-time [<vlan_id>]
```

Task: Remove Dynamic Address Entries

```
# clear mac address-table dynamic [<mac-address>]
```

1.6.2. MAC Address change Notification Traps

- send SNMP trap when the switch learns or removes dynamic and secure MAC addresses.
- do not send trap for self addesses, multicast addresses or static addresses
- can set a trap-interval time to bundle the notification traps to reduce network traffic

Task: Send MAC address change notification traps to an NMS host

```
(config)# snmp-server host <host-addr> { traps | informs} { version { 1 | 2c | 3 } }
<community-string> mac-notification
(config)# snmp-server enable traps mac-notification change
(config)# mac address-table notification change [ interval <seconds> ] [ history-size
<i0-1-500> ]
(config)# interface <interface-id>
(config-if)# snmp trap mac-notification change {added | removed }
```

Task: Verify the MAC address table notification change configuration

```
# sh mac address-table notification change [interface]
```

1.6.3. MAC address move Notification traps

- send a SNMP notification whenever a MAC address moves from one port to another within the same VLAN

Task: Send MAC address move notification traps to an NMS host

```
(config)# snmp-server host <host-addr> { traps | informs} { version { 1 | 2c | 3 } }
<community-string> mac-notification
(config)# snmp-server enable traps mac-notification move
(config)# mac address-table notification mac-move
```

Task: Verify the MAC address table notification move configuration

```
# sh mac address-table notification mac-move
```

1.6.4. MAC Treshold notification traps

- Send an SNMP notification when a MAC Address table treshold limit is reached or exceeded.

Task: Configure MAC Treshold notifcation traps

```
(config)# snmp-server host <host-addr> { traps | informs} { version { 1 | 2c | 3 } }
<community-string> mac-notification
(config)# snmp-server enable traps mac-notification treshold
(config)# mac address-table notification treshold ! to enable the feature
(config)# mac address-table notification treshold [limit <percentage] | [ interval
<seconds> ]
```

Task: Verify the MAC address table notification treshold configuration

```
# sh mac address-table notification treshold
```

1.6.5. Static addresses

- manually entered in the address table and must be manually removed
- can be unicast or mcast
- doesn't age and is retained when the switch restarts
- must be associated with a VLAN and a interface
  - A packet with a static address that arrives on a VLAN where it has not been statically entered is flooded to all ports and not learned
  - if the VLAN is in a private-primary or private-secondary, configure the same static address in all associated VLANs.

Task: Add a static address to the MAC address table

```
(config)# mac address-table static <MAC> vlan <vlan-id> interface <interface-id>
```

Task: Display only static MAC addresses

```
# sh mac address-table static
```

1.6.6. Unicast MAC address filtering

- Drops packets with specific source or destination MAC addressess
- disabled by default
- mcast, bcast and router MAC addresses are not supported

Task: Enable unicast MAC address filtering

```
(config)# mac address-table static <MAC> vlan <vlan-id> drop
```

1.6.7. MAC Address learning on a VLAN

- enabled by default on al VLANs
- can be disabled with the following restrictions:

- If the VLAN has a configured switch virtual interface (SVI), the switch then floods all IP packets in the Layer 2 domain.
- If you disable MAC address learning on a VLAN with more than two ports, every packet entering the switch is flooded in that VLAN domain.
- You cannot disable MAC address learning on a VLAN that is used internally by the switch. If the VLAN ID that you enter is an internal VLAN, the switch generates an error message and rejects the command. To view internal VLANs in use, enter the show vlan internal usage privileged EXEC command.
- If you disable MAC address learning on a VLAN configured as a private-VLAN primary VLAN, MAC addresses are still learned on the secondary VLAN that belongs to the private VLAN and are then replicated on the primary VLAN. If you disable MAC address learning on the secondary VLAN, but not the primary VLAN of a private VLAN, MAC address learning occurs on the primary VLAN and is replicated on the secondary VLAN.
- You cannot disable MAC address learning on an RSPAN VLAN. The configuration is not allowed.
- If you disable MAC address learning on a VLAN that includes a secure port, MAC address learning is not disabled on that port. If you disable port security, the configured MAC address learning state is enabled.

Task: Disable MAC address learning

```
(config)# no mac address-table learning vlan <vlan-id>
```

Task: Display the MAC address learning

```
sh mac address-table learning [vlan <vlan-id>]
```

Task: Reenable MAC address learning

```
(config)# default mac address-table learning vlan <vlan-id>
```

1.7. errdisable recovery

1.8. L2 MTU

Chapter 2. Ethernet

- IEEE 802.3 standards
- CSMA/CD protocol
- Medium: coaxial, twisted-pair, optical fiber
- Data rates: 10/100/1000/10000 Mbps

2.1. Frame Formats

8	6	6	2	46-1500	4
Preamble	DA	SA	Type/Length	Data and Padding	FCS

Figure 1. Ethernet (DIX) and Revised (1997) IEEE 802.3

8	6	6	2	1	1	1-2	43-1500	4
Preamble	DA	SA	Length	DSAP	SSAP	Control	Data and Padding	FCS

Figure 2. Original IEEE 802.3

8	6	6	2	1	1	1-2	3	2	43-1500	4
Preamble	DA	SA	Length	DSAP	SSAP	Control	OUI	Type	Data and Padding	FCS

Figure 3. IEEE 802.3 with SNAP

Preamble DIX or Preamble and Start of Frame Delimiter(802.3)

- 62 alternating 1s and 0s, and ends with a pair of 1s.
- For clocking synchronization of the transmitted signal.

Type

- Type of protocol

Length

- Length in bytes of the data following the Length field, up to the Ethernet trailer.

DA

- Destination address can be an individual or group address

SA

- Source address is always unicast address

DSAP

- Destination Service Access Protocol
- The size limitations, along with other Point (802.2) uses of the low-order bits, required the later addition of SNAP headers.

SSAP

- Source Service Access Protocol
- Describes the upper-layer protocol Point (802.2) that created the frame.

Control

- Enables both connectionless and connection-oriented operation.
- Generally used only for connectionless operation by modern protocols, with a 1-byte value of 0x03.

SNAP OUI

- Generally unused today,
- Providing a place for the sender of the frame to code the OUI representing the manufacturer of the Ethernet NIC.

SNAP Type

- Using same values as the DIX Type field, overcoming deficiencies with size and use of the DSAP field.

Data

- N bytes where 46 ⇐ n ⇐ 1500
- If n < 46, use padding

FCS (Frame check sequence)

- Contains a 32-bit cyclic redundancy check (CRC) value
- Calculated by the sending MAC
- Re-calculated by the receiving MAC to check for damaged frames.
- Generated from the DA, SA, Length/Type, and Data fields

2.2. Ethernet MAC Addresses

- 48 bits in hexadecimal
- Canonical transmission (little endian)= MSO to LSO with LSB to MSB for each octet where
  - I/G bit: (0/1) Individual or Group address, first bit to be transmitted as LSB of MSO
  - U/L bit: (0/1) Universally or Locally administrated, second bit to be transmitted

```
Example: AC-10-7B-3A-92-3C

Convert to Hexa : 10101100 00010000 01101011 00111010 01010010 00111100
Transmission    : 00110101 00001000 11010110 01011100 01001010 00111100
```

Task: Change the MAC address

```
(config-if)# mac-address AC-10-BE-EF-DE-AD
```

Even if you change the MAC address of the swithch port, STP will continue to use the BIA.

2.2.1. Types of MAC addresses

- Unicast : I/G bit = 0
- Multicast: I/G bit = 1
- Broadcast: all devices in the segment

2.3. RJ-45 pinouts and Cat5 wiring

- Defined by EIA / TIA

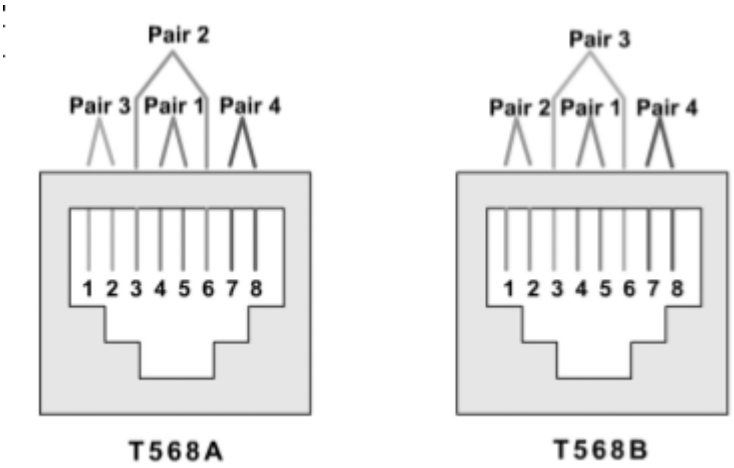


Table 1. Ethernet cabling types

Type of cable	Pinouts	Key pins connected
Straight-through	T568A or T568B both ends	1-1; 2-2; 3-3; 6-6
Cross-over	T568A on one end, T568B on the other	1-3; 2-6; 3-1; 6-2

- Auto-MDIX (automatic medium-dependent interface crossover)
  - Detects the wrong cable and causes the switch to swap the pair it uses for transmitting and receiving
  - Not supported on all Cisco switch models

2.4. Auto-negotiation, Speed and Duplex

- By default, Ethernet auto-negotiation uses FLP (Fast Link Pulses)to determine the speed and duplex setting.

- To disable auto-negotiation, manually configure the speed and the duplex settings.
- if auto-negotation is disabled on one end by statically setting the speed , the other end
  - detects the speed based on the incoming electrical signal
  - sets duplex to half for 10 and 100 Mbps and full duplex for 1Gps interfaces
- if auto-negotation is disabled on both end and different speeds statically configured, link down

Task: Set speed for the interface

```
(config-if)# speed {10 | 100 | 1000 | auto | nonegotiate}
```

Task: Set duplex mode for the interface

```
(config-if)# duplex {auto | full | half}
```

Task: Show controllers

```
Router# show controllers fastethernet1
!
Interface FastEthernet1   MARVELL 88E6052
Link is DOWN
Port is undergoing Negotiation or Link down
Speed :Not set, Duplex :Not set
!
Switch PHY Registers:
~~~~~~~~~~~~~~~~~~~~~
00 : 3100   01 : 7849   02 : 0141   03 : 0C1F   04 : 01E1
05 : 0000   06 : 0004   07 : 2001   08 : 0000   16 : 0130
17 : 0002   18 : 0000   19 : 0040   20 : 0000   21 : 0000
!
Switch Port Registers:
~~~~~~~~~~~~~~~~~~~~~~
Port Status Register       [00] : 0800
Switch Identifier Register [03] : 0520
Port Control Register      [04] : 007F
Rx Counter Register        [16] : 000A
Tx Counter Register        [17] : 0008
```

2.5. Switch internal processing

Switches forward frames when necessary, and do not forward when there is no need to do so, thus reducing overhead.

To accomplish this, switches perform three actions:

- Learn MAC addresses by examining the source MAC address of each received frame
- Decide when to forward a frame or when to filter (not forward) a frame, based on the

destination MAC address

- Create a loop-free environment with other bridges by using the Spanning Tree Protocol

Store-and-forward

The switch fully receives all bits in the frame (store) before forwarding the frame (forward). This allows the switch to check the FCS before forwarding the frame, thus ensuring that errored frames are not forwarded.

Cut-through

The switch performs the address table lookup as soon as the Destination Address field in the header is received. The first bits in the frame can be sent out the outbound port before the final bits in the incoming frame are received. This does not allow the switch to discard frames that fail the FCS check, but the forwarding action is faster, resulting in lower latency.

Fragment-free

This performs like cut-through switching, but the switch waits for 64 bytes to be received before forwarding the first bytes of the outgoing frame. According to Ethernet specifications, collisions should be detected during the first 64 bytes of the frame, so frames that are in error because of a collision will not be forwarded.

2.6. Switching and bridging logic

Type of Address	Switch Action
Known unicast	Forwards frame out the single interface associated with the destination address
Unknown unicast	Floods frame out all interfaces, except the interface on which the frame was received
Broadcast	Floods frame identically to unknown unicasts
Multicast	Floods frame identically to unknown unicasts, unless multicast optimizations are configured

2.7. Standards

802.1Q	dot1q trunking
802.1d	STP
802.1s	MST
802.1w	Rapid STP
802.1ax	LACP (formerly 802.3ad)
802.2	Logical Link Control
802.3u	Fast ethernet over copper and optical cable
802.3z	Gigabit ethernet over optical cable

802.3ab Gigabit ethernet over copper cable

Table 1-12 *Ethernet Types and Cabling Standards*

Standard	Cabling	Maximum Single Cable Length
10BASE5	Thick coaxial	500 m
10BASE2	Thin coaxial	185 m
10BASE-T	UTP Cat 3, 4, 5, 5e, 6	100 m
100BASE-FX	Two strands, multimode	400 m
100BASE-T	UTP Cat 3, 4, 5, 5e, 6, 2 pair	100 m
100BASE-T4	UTP Cat 3, 4, 5, 5e, 6, 4 pair	100 m
100BASE-TX	UTP Cat 3, 4, 5, 5e, 6, or STP, 2 pair	100 m
1000BASE-LX	Long-wavelength laser, MM or SM fiber	10 km (SM) 3 km (MM)
1000BASE-SX	Short-wavelength laser, MM fiber	220 m with 62.5-micron fiber; 550 m with 50-micron fiber
1000BASE-ZX	Extended wavelength, SM fiber	100 km
1000BASE-CS	STP, 2 pair	25 m
1000BASE-T	UTP Cat 5, 5e, 6, 4 pair	100 m

2.8. Troubleshooting

- Add something about excessive collisions, late collisions, runts, re: duplex mismatches

Runts

- Runts are frames smaller than 64 bytes.

CRC errors

- The frame's cyclic redundancy checksum value does not match the one calculated by the switch or router.

Frames

- Frame errors have a CRC error and contain a noninteger number of octets.

Alignment

- Alignment errors have a CRC error and an odd number of octets.

Collisions

- Look for collisions on a full-duplex interface (meaning that the interface operated in half-duplex mode at some point in the past), or excessive collisions on a half-duplex interface.

Late collisions on a half-duplex interface

- A late collision occurs after the first 64 bytes of a frame.

2.8.1. Problem and approches

Problem	Questions?	Commands
Lack of reachability to devices in the same VLAN	- Layer 1 issues ? - VLAN exists on the switch? - Interface assigned to the correct VLAN? - VLAN allowed on the trunk?	- show interface - show vlan - show interface switchport - traceroute mac source-mac destination-mac - show interface trunk
Intermittent reachability to devices in the same VLAN	- Excessive interface traffic? - Unidirectional links? - Spanning-tree problems such as BPDU floods or flapping MAC addresses?	- show interface - show spanning-tree show spanning-tree root show mac address-table
No connectivity between switches	- Trunk links active? - EtherChannels active? - BPDU Guard is not enabled on a trunk interface?	- show interfaces status err-disabled - show interfaces trunk - show etherchannel summary - show spanning-tree detail
Poor performance across a link	- Duplex mismatch?	- show interface

Chapter 3. CDP, LLDP and UDLD

3.1. CDP

Catalyst3560-X Configuration Guides | **CDP**

3.1.1. Overview

- Layer 2 discovery protocol running on Cisco devices
- Retrieves device type and SNMP agent address of neighboring devices

Packet format

- Header followed by a set of TLV value

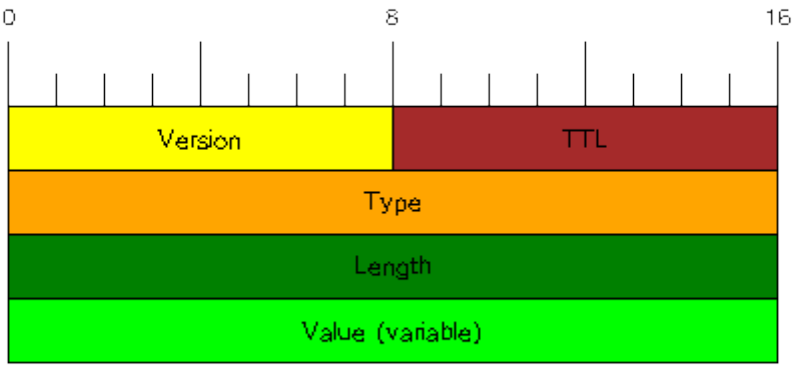


Figure 4. CDP frame format

3.1.2. CDP operations

- Enable by default

Task: Display global information about CDP characteristics

```
# show cdp

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
Router3             Ser 1          120           R        2500      Ser 0
Router1             Eth 1          180           R        2500      Eth 0
Switch1             Eth 0          240           S        1900      2
```

```
show cdp entry <entry-name> [protocol | version]
```

Task: Disable CDP

```
(config)# no cdp run
```

Task: Enable CDP on an interface

```
(config-if)# cdp enable
```

CDP updates

Task: Set the transmission frequency of CDP updates in seconds

```
(config)# cdp timer <seconds>
```

Default: 60 seconds range: 5-254 seconds

Task: Specify the amount of time a receiving device should hold the information sent by your device

```
(config)# cdp holdtime <seconds>
```

default: 180 seconds, range: 10 to 255 seconds

Version

Task: Send Version-2 advertisements

```
(config)# cdp advertise-v2
```

3.1.3. Monitoring and maintenance

Task: Reset the traffic counters to zero

```
clear cdp counters
```

Neighbors

Task: Delete the CDP table of information about neighbors

```
clear cdp table
```

Task: Display information about interfaces where CDP enabled

```
sh cdp interface [<interface-id>]
```

Task: Display information about neighbors

```
sh cdp neighbors [<interface-id>] [detail]
```

Task: Display CDP counters, including the number of packets sent and received and checksum errors

```
# show cdp traffic

Total packets output: 543, Input: 333
 Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
 No memory: 0, Invalid: 0, Fragmented: 0
 CDP version 1 advertisements output: 191, Input: 187
 CDP version 2 advertisements output: 352, Input: 146
```

3.2. LLDP

Catalyst Configuration Guides | **LLDP**

3.2.1. Overview

- IEEE 802.1AB link layer discovery protocol
- neighbor discovery protocol
- advertises TLV(type, length, value) for each attribute
  - basic mandatory
    - port description
    - system name
    - system description
    - system capabilities
    - management address
  - optional
    - port vlan ID for ieee 802.1
    - MAC/PHY configuration/status for ieee 802.3

3.2.2. LLDP global state

- Disabled by default

Task: Enable LLDP globally on the switch

```
(config)# lldp run
```

Task: Display global information, such as frequency of transmissions, the holdtime for packets being sent, and the delay time before LLDP initializes on an interface.

```
show lldp
```

Task: Display LLDP counters, including the number of packets sent and received, number of packets discarded, and number of unrecognized TLVs.

```
show lldp traffic
```

Task: Reset the traffic counters to zero.

```
clear lldp counters
```

Task: Delete the LLDP neighbor information table.

```
clear lldp table
```

Task: Clear the NMSP statistic counters.

```
clear nmsp statistics
```

3.2.3. LLDP interfaces

- Disabled by default

Task: Enable an interface to send LLDP packets

```
(config-if)# lldp transmit
```

Task: Enable an interface to receive LLDP packets

```
(config-if)# lldp receive
```

Task: Display information about interfaces with LLDP enabled.

```
show lldp interface [<interface-id>]
```

3.2.4. Neighbors

Task: Display information about a specific neighbor.

```
show lldp entry <entry-name>
```

Task: Display information about all neighbors.

```
show lldp entry *
```

Task: Display information about neighbors, including device type, interface type and number, holdtime settings, capabilities, and port ID.

```
show lldp neighbors [<interface-id>] [detail]
```

3.2.5. Timers

Task: Specify the amount of time a receiving device should hold the information from your device

- default: 120 s, range: 0 - 65535

```
(config)# lldp holdtime <seconds>
```

Task: Specify the delay time in seconds for LLDP to initialize on an interface.

The range is 2 to 5 seconds; the default is 2 seconds.

```
(config)# lldp reinit delay
```

Task: Set the sending frequency of LLDP updates in seconds.

The range is 5 to 65534 seconds; the default is 30 seconds.

```
(config)# lldp timer rate
```

3.2.6. TLV

Task: Specify the LLDP TLVs to send or receive.

```
(config)# lldp tlv-select
```

Task: Specify the LLDP-MED TLVs to send or receive.

```
(config)# lldp med-tlv-select
```

Task: Specify the LLDP-MED TLV to send

```
(config-if)# lldp med-tlv-select {inventory-management | location | network-policy |
power-management }
```

Task: Configure network policy TLV

```
(config)# network-policy profile <profile-number>
(config)#  {voice | voice-signaling} vlan [<id>  {cos <cvalue> | dscp <dvalue>}]
            | [[dot1p {cos <cvalue> | dscp <dvalue>}] | none | untagged]
(config-if)# network-policy <profile-number>
(config-if)# lldp med-tlv select network-policy
```

- if the interface is configured as a tunnel port, LLDP is automatically disabled.
- If you first configure a network-policy profile on an interface, you cannot apply the switchport voice vlan command on the interface. If the switchport voice vlan vlan-id is already configured on an interface, you can apply a network-policy profile on the interface. This way the interface has the voice or voice-signaling VLAN network-policy profile applied on the interface.
- You cannot configure static secure MAC addresses on an interface that has a network-policy profile.
- You cannot configure a network-policy profile on a private-VLAN port.
- For wired location to function, you must first enter the ip device tracking global configuration command.

Task: Display the location information for an endpoint.

```
show location
```

3.2.7. Network-policy profiles

Task: Display the configured network-policy profiles.

```
show network-policy profile
```

Task: Display the NMSP information.

```
show nmsp
```

3.2.8. LLDP-MED

- LLDP for Media Endpoint Devices
- operates between endpoint devices (ip phones) and network devices (switches)
- supports VoIP applications
- TLVs enabled by default:
  - LLDP-MED capabilities TLV
  - network policy TLV

- Power management TLV
- Inventory management TLV
- Location TLV

3.2.9. Wired location service

- The switch uses the wired location service feature to send location and attachment tracking information for its connected devices to a Cisco Mobility Services Engine (MSE). The tracked device can be a wireless endpoint, a wired endpoint, or a wired switch or controller. The switch notifies the MSE of device link up and link down events through the Network Mobility Services Protocol (NMSP) location and attachment notifications.

The MSE starts the NMSP connection to the switch, which opens a server port. When the MSE connects to the switch there are a set of message exchanges to establish version compatibility and service exchange information followed by location information synchronization. After connection, the switch periodically sends location and attachment notifications to the MSE. Any link up or link down events detected during an interval are aggregated and sent at the end of the interval.

When the switch determines the presence or absence of a device on a link-up or link-down event, it obtains the client-specific information such as the MAC address, IP address, and username. If the client is LLDP-MED- or CDP-capable, the switch obtains the serial number and UDI through the LLDP-MED location TLV or CDP.

Depending on the device capabilities, the switch obtains this client information at link up:

- Slot and port specified in port connection
- MAC address specified in the client MAC address
- IP address specified in port connection
- 802.1X username if applicable
- Device category is specified as a wired station
- State is specified as new
- Serial number, UDI
- Model number
- Time in seconds since the switch detected the association

Depending on the device capabilities, the switch obtains this client information at link down:

- Slot and port that was disconnected
- MAC address
- IP address
- 802.1X username if applicable
- Device category is specified as a wired station
- State is specified as delete

- Serial number, UDI
- Time in seconds since the switch detected the disassociation

When the switch shuts down, it sends an attachment notification with the state delete and the IP address before closing the NMSP connection to the MSE. The MSE interprets this notification as disassociation for all the wired clients associated with the switch.

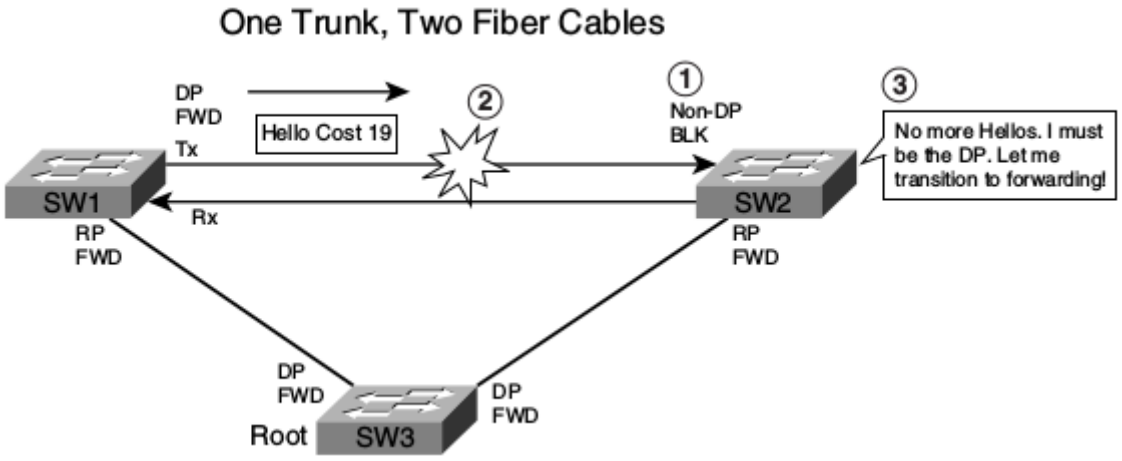
If you change a location address on the switch, the switch sends an NMSP location notification message that identifies the affected ports and the changed address information.

3.3. UDLD

Catalyst 3560 Configuration Guides | **UDLD**

3.3.1. Overview

- Problem: unidirectional links
  - one of the 2 transmission paths has failed but not both
  - due to miscabling, cutting on fiber cable, unplugging one fiber, GBIC problems, ...
  - can cause a loop as the previously blocking port will move to a forwarding state



- solutions:

  UDLD **u***ni***d***irectional* **l***ink* **d***etection*

  Uses Layer 2 messaging to decide when a switch can no longer receive frames from a neighbor. The switch whose transmit interface did not fail is placed into an err-disabled state.

  UDLD aggressive mode

  Attempts to reconnect with the other switch (eight times) after realizing no messages have been received. If the other switch does not reply to the repeated additional messages, both sides become err-disabled.

Modes of operations

Normal

- default
- detects unidirectional links due to misconnected ports on fiber-optic connection

Aggresive mode

3.3.2. Tasks

Default configuration

Feature	Default Setting
UDLD global enable state	Globally disabled
UDLD per-port enable state for fiber-optic media	Disabled on all Ethernet fiber-optic ports
UDLD per-port enable state for UTP copper media	Disabled on all Ethernet 10/100/1000BASE-TX ports
UDLD aggressive mode	Disabled

Task: Enable UDLD globally

```
(config)# udld {aggressive | enable | message time <seconds>}
```

message time <seconds>

- configure the period of time between UDLD probe messages on ports that are in the advertisement phase and are detected to be bidirectional.
- range: 1 to 90 seconds
- default: 15 seconds
- This command affects fiber-optic ports only. Use **(config-if)# udld** to enable UDLD on other port types.

Task: Reset an interface disabled by UDLD

```
udld reset
```

You can also restart the disabled port

- **shutdown** followed by **no shutdown**
- **no udld {aggressive** | **enable}** followed by **udld {aggressive** | **enable}**
- **no udld port** followed by **udld port [aggressive]**

3.3.3. UDLD error-disabled state

Task: Recover from the UDLD error-disabled state

```
! Enable UDLD to automatically recover
(config)# errdisable recovery cause udld

! Specify the time to recover from the UDLD error-disabled state
(config-if)# errdisable recovery interval <seconds>
```

Task: Display UDLD status

```
show udld [interface-id]
```

Chapter 4. VLANs and Trunking

4.1. Normal and Extended VLANs

- Administratively defined subset of switch ports that are in the same broadcast domain
- Best practice: one VLAN, one IP subnet
- Traffic inside same VLAN is layer 2 switched
- Traffic between VLANs is layer 3 routed
- Can span multiple physical switches over "trunks"

4.1.1. VLAN numbering

- VLAN ID = 12 bits

  Reserved [0, 4095]
    - Not available for use

  Normal-range [1-1005]
    - Advertised and pruned by VTP v1 and v2 except vlan 1, 1002-1005
    - Configured in both vlan database mode and configuration mode
    - Stored in VLAN.DAT file in Flash
    - Special VLANs:
      - Vlan 1 is the default Ethernet VLAN for all access ports; cannot be deleted or changed.
      - Vlan 1002-1004 : default for FDDI
      - Vlan 1002-1005 : default for Token Ring translational bridge.

  Extended-range [1006-4094]
    - Cannot be advertised or pruned by VTP v1 and v2
    - Configured only in VTP transparent mode
    - Stored only in the running configuration

4.1.2. VLAN Trunks

- Trunk: point-to-point links for multiple VLANs between devices
- Trunking add ISL or 802.1q headers to include VLAN id.
  - ISL : Cisco proprietary, 30-bytes (26-byte header + 4-byte trailer), does not modify original frame
  - 802.1q: IEEE standard, 4-byte tag except for native VLAN, modifies original frame

4.1.3. Basic configuration

Configuring VLANs requires few steps:

1. Create the VLAN itself
2. Associate the correct ports with VLAN

VLAN creation can be done either in VLAN database mode configuration (after **vlan database**) or normal configuration mode

Table 2. Catalyst 3550 VLAN database vs configuration mode

VLAN Database	Configuration
vtp {domain domain-name | password password | pruning | v2-mode | {server | client | transparent}}	vtp {domain domain-name | file filename | interface name | mode {client | server | transparent} | password password | pruning | version number}
vlan vlan-id [backupcrf {enable | disable}] [mtu mtu-size] [name vlan-name] [parent parent-vlan-id] [state {suspend | active}]	vlan vlan-id 1
show {current | proposed | difference}	No equivalent
apply | abort | reset	No equivalent

4.1.4. VLAN state

- Can be active or suspended

Task: Modify the operational state of a VLAN

```
(config)# vlan <id>
(config-vlan)#state [active | suspend]
```

4.1.5. Troubleshoot

Check "Creating ethernet VLANs on catalyst switches: troubleshoot tips"

- SVI will be in "up/down" state after being deleted
- SVI will be in "up/up" if
  - The VLAN associated with the SVI exists in the VLAN database
  - At least one trunk or access port in the "up/up" state has been assigned to the VLAN
  - Those ports in the "up/up" state are not blocked by STP

4.2. Voice VLANs

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-

0_2_se/configuration/guide/3750x_cg/swvoip.html

4.3. Private VLANs

Cat3560-X Configuration Guides | Private VLANs

- Separate ports as if they are on different VLAN while consuming only one subset.
- Typically used by service provider in a multi-tenant offerings: one router, one switch, multiple customers
- PVLAN
  - one primary VLAN (promiscuous ports) + multiple secondary VLANs
  - At most one isolated secondary VLAN
  - Zero or more community secondary VLANs
- Cannot use vlan 1, 1002-1005 as private vlans

Task: Display private VLAN information

```
# sh vlan private-vlan [type]

Primary     Secondary   Type               Ports
-------     ---------   ----               ---------------------
10          501         isolated           Gi2/0/1,Gi3/0/1,Gi3/0/2
10          502         community          Gi2/0/11,Gi3/0/1,Gi3/0/4
10          503         non-operational
```

4.3.1. Primary VLANs

- carries unidirectional traffic downstream from promiscuous ports to all ports

Task: Configure primary VLAN and associated secondary VLANs

```
(config)# vtp mode transparent
(config)# vlan 100
(config-vlan)#private-vlan primary
(config-vlan)#private-vlan association 123-321,999
```

4.3.2. Isolated VLANs

- carries unidirectional traffic upstream from the hosts toward the promiscuous ports and the gateway.

Task: Configure isolated VLANs

```
(config)# vtp mode transparent
(config)# vlan 102
(config-vlan)# private-vlan isolated
```

4.3.3. Community VLANs

- carries unidirectional traffic upstream from the hosts toward the promiscuous ports and the gateway.

Task: Configure isolated VLANs

```
(config)# vtp mode transparent
(config)# vlan 102
(config-vlan)# private-vlan isolated
```

4.3.4. Private-Vlan Host port

Task: Configure a layer 2 port interface as a private-vlan host port

```
(config-if)# switchport mode private-vlan host
(config-if)# switchport private-vlan host-association <primary-vlan-id> <secondary-
vlan-ids>
```

NOTE	Although private VLANs provide host isolation at Layer 2, hosts can communicate with each other at Layer 3.

4.3.5. Private-VLAN promiscuous ports

Task: Configure a layer 2 port interface as a private-vlan promiscuous port

```
(config-if)# switchport mode private-vlan promiscuous
(config-if)# switchport private-vlan mapping <primary-vlan-id> [add | remove]
<secondary-vlan-ids>
```

4.3.6. Private-VLAN SVI

- SVI can only be associated to primary VLANs
- SVIs for secondary VLANs are inactive
- If you assign an IP subnet to the primary VLAN SVI, this subnet is the IP subnet address of the entire private VLAN

Task: Configure a layer 3 vlan interface as a private-vlan promiscuous port

```
(config)# interface vlan <primary-vlan-id>
(config-if)# private-vlan mapping <primary-vlan-id> [add | remove] <secondary-vlan-
ids>
```

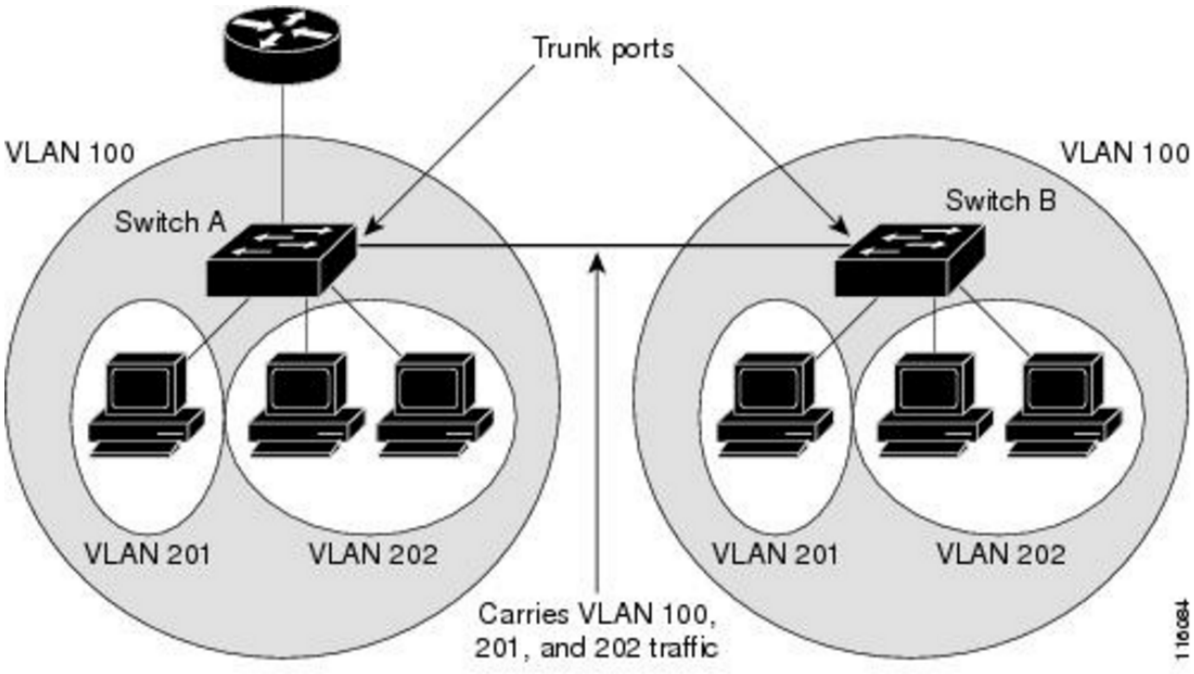
Task: Display information about the private-VLAN mapping for VLAN SVIs

```
# sh interface private-vlan mapping
```

TODO ip sticky arp

4.3.7. Private-vlan accross multiple switches

As with regular VLANs, private VLANs can span multiple switches. A trunk port carries the primary VLAN and secondary VLANs to a neighboring switch. The trunk port treats the private VLAN as any other VLAN. A feature of private VLANs across multiple switches is that traffic from an isolated port in switch A does not reach an isolated port on Switch B.



VLAN 100 = Primary VLAN
VLAN 201 = Secondary isolated VLAN
VLAN 202 = Secondary community VLAN

WARNING	• Manually configure private VLANs on all switches because VTP (v1 and v2) does not support private VLANs, -

TODO interaction with switch that do not support private-vlan

TODO PVLAN Trunk

TODO PVLAN Isolated

TODO see page 67 Narbick

4.3.8. Interaction with other features

VTP

- VTP v1 andv2 don't propagate private-vlans
  - Set transparent mode
  - Save the VTP transparent mode and private-vlan to startup configuration
- VTP v3 supports private-vlans

STP

- only one STP instance for the entire private-vlan
- the STP parameters of the primary VLAN are propagated to the secondary VLANs
- Enable Port Fast and BPDU guard on isolated and community host ports to prevent STP loops due to misconfigurations and to speed up STP convergence
- Do not enable Port Fast and BPDU guard on promiscuous ports.

DHCP snooping

- Can be enabled on the private VLAN
- propagates to all secondary vlans when enabled on the primary VLAN
- If you configure DHCP on a secondary VLAN, the configuration does not take effect if the primary VLAN is already configured (?!)

IP source guard

- enabled only if DHCP snooping is enabled on the primary vlan

SPAN

- You can configure a private-VLAN port as a SPAN source port.
- You can use VLAN-based SPAN (VSPAN) on primary, isolated, and community VLANs or use SPAN on only one VLAN to separately monitor egress or ingress traffic.
- A private-VLAN host or promiscuous port cannot be a SPAN destination port. If you configure a SPAN destination port as a private-VLAN port, the port becomes inactive.
- A RSPAN vlan can not be a private-vlan primary or secondary vlan.

PAgP or LACP

- If a port is part of a private vlan, any Etherchannel configuration is inactive

IGMP snooping

- When enabled (the default), the switch supports no more than 20 private-vlan domain

802.1x

- You can configure IEEE 802.1x port-based authentication on a private-VLAN port,

- You can not configure IEEE 802.1x with port security, voice VLAN, or per-user ACL on private-VLAN ports.

Static MAC address

- If you configure a static MAC address on a promiscuous port in the primary VLAN, you must add the same static address to all associated secondary VLANs.
- If you configure a static MAC address on a host port in a secondary VLAN, you must add the same static MAC address to the associated primary VLAN.
- When you delete a static MAC address from a private-VLAN port, you must remove all instances of the configured MAC address from the private VLAN.

4.4. VTP

Catalyst Configuration guides | **Configuring VTP**

- Vlan Trunk Protocol
- Cisco-proprietary that distributes VLAN information among Catalyst switches
- Advertises the VLAN Id, Name and Type but not which ports should be in each VLAN
- Eases administrative burden for addition, deletion and renaming of VLANs
- Supports 1005 VLANs (IP base or IP services feature set) or 255 VLANs (LAN base feature set)

Task: Show VTP status

```
# show vtp status

VTP Version: 3 (capable)
Configuration Revision: 1
Maximum VLANs supported locally: 1005
Number of existing VLANs: 37
VTP Operating Mode: Server
VTP Domain Name: [smartports]
VTP Pruning Mode: Disabled
VTP V2 Mode: Enabled
VTP Traps Generation: Disabled
MD5 digest : 0x26 0xEE 0x0D 0x84 0x73 0x0E 0x1B 0x69
Configuration last modified by 172.20.52.19 at 7-25-08 14:33:43
Local updater ID is 172.20.52.19 on interface Gi5/2 (first layer3 interface fou)
VTP version running: 2
```

4.4.1. VTP Version

V1

- Default: version 1
- supports normal range only

V2

- supports for Token Ring Concentrator Relay Function and Bridge Relay Function
- propagates unknown TLV records
- Optimized VLAN database consistency checking:
  - In VTPv1, VLAN database consistency checks are performed whenever the VLAN database is modified, either through CLI, SNMP, or VTP.
  - In VTPv2, these consistency checks are skipped if the change was caused by a received VTP message, as the message itself was originated as a result of a CLI or SNMP action that must already have been sanitized.
  - This is really just an implementation optimization.

V3

- Supports the whole IEEE 802.1q vlan range up to 4095 (v1 and v2 support only normal range VLANs 1-1005)
- Can send private LAN information in addition to normal VLAN information.
- Backward compatible with VTP 2
- Add support for databases other than VLAN databases such as MST databases.
- Clear text or hidden password protection
  - The encrypted VTP password cannot be displayed back as plaintext.
  - While this encrypted string can be carried over to a different switch to make it a valid member of the domain, the promotion of a secondary server into the primary server role will require entering the password in its plaintext form.
- Supports the **off** mode in which the switch does not participate in VTPv3 operations and drops all received VTP messages:
- Can deactivate VTP on a per-trunk basis
- VTPv3 is a generalized mechanism for distributing contents of an arbitrary database, and is not limited to synchronizing VLAN information over a set of switches:
  - As an example, VTPv3 is also capable of distributing and synchronizing the MST region configuration among all switches in a VTP domain

For more information, read VTP version 3

4.4.2. VTP message format

- Encapsulated in ISL or 802.1q frames
- Multicasted to MAC address: 0100-0CCC-CCCC, LLC code: SNAP (AAAA), Type 2003 in the SNAP Header
- Carried through trunk ports and VLAN 1

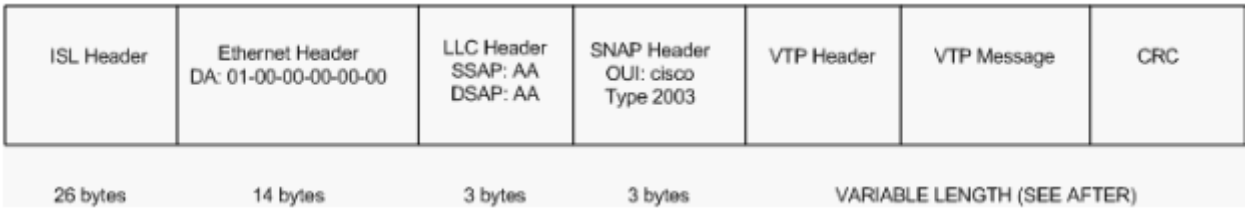


Figure 5. Example: VTP packet encapsulated in ISL frame

- The VTP header contains these fields:
  - VTP protocol version: 1,2,3
  - VTP message types: summary advertisements, subset advertisements, advertisement requests, VTP join messages
  - management domain length
  - management domain name

Summary advertisements

- Sent by Server and Client every 5 minutes intervals, and in addition, after each modificaton of the VLAN database
- carries information about VTP domain name, revision number, identity of the last updater, time stamp of the last update, MD5 sum computed over the contents of the VLAN database and the VTP password (if configured), and the number of Subset Advertisement messages that optionally follow this Summary Advertisement.
- summary messages do not carry VLAN database contents.
- When the switch receives a summary advertisement packet,
  - The switch compares the VTP domain name to its own VTP domain name.
  - If the name is different, the switch simply ignores the packet.
  - If the name is the same, the switch then compares the configuration revision to its own revision.
  - If its own configuration revision is higher or equal, the packet is ignored.
  - If it is lower, an advertisement request is sent.

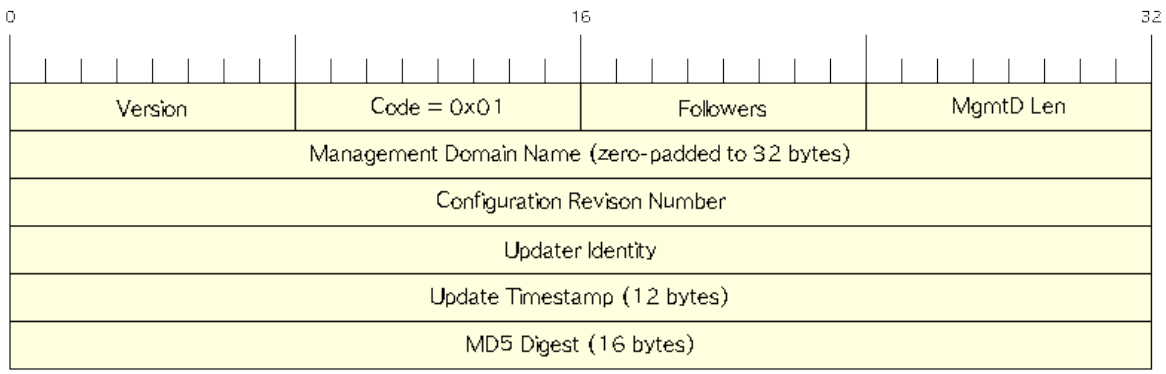


Figure 6. VTP Summary Advertisement

Field	Description
Followers	Indicates that this packet is followed by a Subset Advertisement packet.
Updater Identity	IP address of the switch that is the last to have incremented the configuration revision.
Update Timestamp	Date and time of the last increment of the configuration revision.
MD5 Digest	If MD5 is configured and used to authenticate the validation of a VTP update.

Subset advertisements

- Follows the summary advertisement after addition, deletion or modification of a VLAN.
- Contains a list of VLAN information.

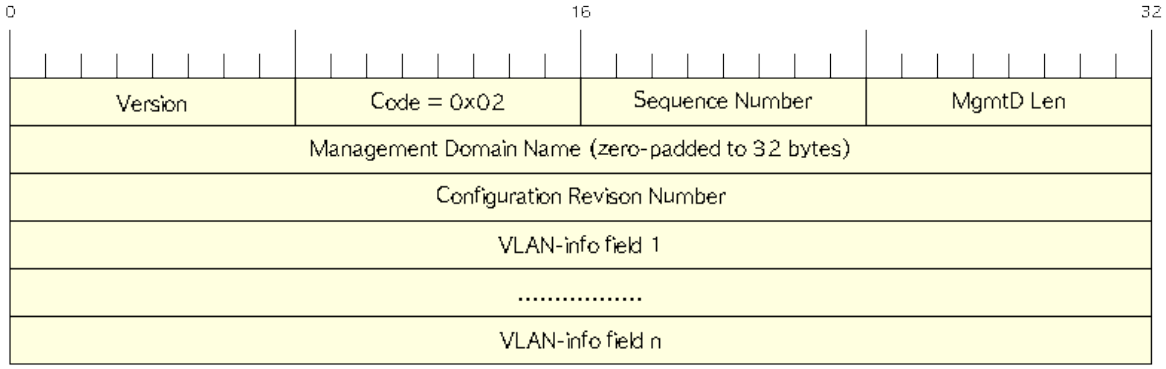


Figure 7. VTP Subset advertisements

Sequence number

- Identify the packet in the stream of packets that follow a summary advertisement
- Starts with value 1

Advertisement request

A switch needs a VTP advertisement request in these situations:

- The switch has been reset.
- The VTP domain name has been changed.
- The switch has received a VTP summary advertisement with a higher configuration revision than its own.

Upon receipt of an advertisement request, a VTP device sends a summary advertisement. One or more subset advertisements follow the summary advertisement. This is an example:

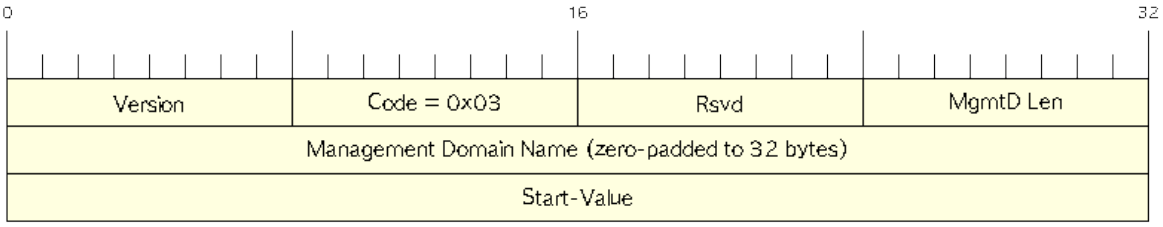


Figure 8. VTP advertisement request

Start-Value

This is used in cases in which there are several subset advertisements. If the first (n) subset advertisement has been received and the subsequent one (n+1) has not been received, the Catalyst only requests advertisements from the (n+1)th one.

Join Message

- originated by each VTP Server and Client switch periodically every 6 seconds if VTP Pruning is active.
- Join messages contain a bit field that, for each VLAN in the normal range, indicates whether it is active or unused (that is, pruned)

IMPORTANT	In any VTP version, VTP messages are transmitted and accepted only on trunk ports. Access ports neither send nor accept VTP messages. For two switches to communicate in VTP, they must first be interconnected through a working trunk link.

4.4.3. VTP domain

- Controls which devices can exchange VTP advertisements
- Defaults to NULL value
- Switch inherits VTP domain name of first received advertisement over trunk links
- A switch can only be part of one domain at a time

Task: Set the VTP domain name

```
(config)# vtp domain <name>
```

4.4.4. Configuration revision number

- 32-bit
- Incremented by one for each configuration change
- Higher revision indicates newer database

IMPORTANT	for a newly connected VTP server or client to change another switch's VTP database, the following must be true: - The new link connecting the new switch is trunking. - The new switch has the same VTP domain name as the other switches. - The new switch's revision number is higher than that of the existing switches. - The new switch must have the same password, if configured on the existing switches.

4.4.5. VTP modes

You can configure a switch to operate in any one of these VTP modes:

Server

- Default mode
- Allows addition, deletion and modification of VLAN information
- Changes on server overwrite the rest of the domain
- Configuration saved in NVRAM

Task: Configure the switch as a VTP server

```
(config)# vtp mode server
```

Client

- Cannot add, remove or modify VLAN information
- Listens for advertisements originated by server, install them and passes them on
- Configuration saved in NVRAM only for VTPv3

Task: Configure the switch as a VTP client

```
(config)# vtp mode client
```

Transparent

- Keeps a separate VTP database from the rest of the domain
- Does not originate advertisements
- "transparently" passes received advertisements through without installing them
- Can still create, remove or renamed VLANs which are not advertised to neighboring switches.
- Need for some applications like Private VLANs

Task: Setup VTP transparent mode

```
(config)# vtp mode transparent
```

Off (configurable only in CatOS switches)

- Like VTP transparent mode with the exception that VTP advertisements are not forwarded

Table 3. VTP Modes and Features

Function	Server Mode	Client Mode	Transparent Mode
Originates VTP advertisements	Yes	Yes	No
Processes received advertisements to update its VLAN configuration	Yes	Yes	No
Forwards received VTP advertisements	Yes	Yes	Yes
Saves VLAN configuration in NVRAM or vlan.dat	Yes	Yes	Yes
Can create, modify, or delete VLANs using configuration commands	Yes	No	Yes

4.4.6. VTP security

- MD5 authentication prevents against certain attack
  - Does not prevent against misconfiguration
  - Password must be setup manually because switches only exchanges MD5 digest of the password.

Task: Configure VTP authentication

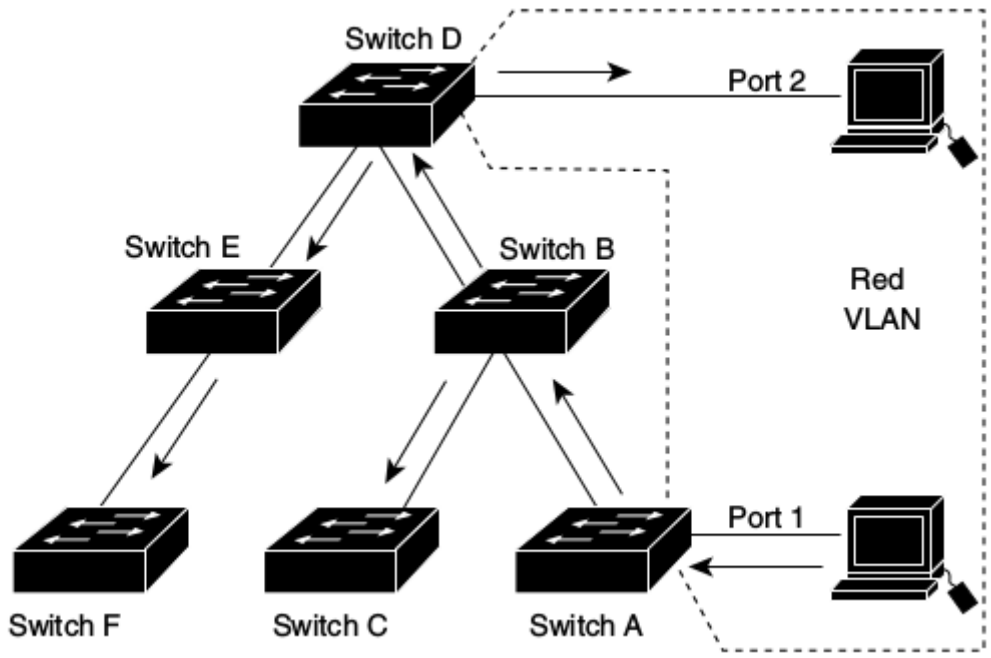
```
(config)# vtp password <string>
```

Task: Show the VTP password

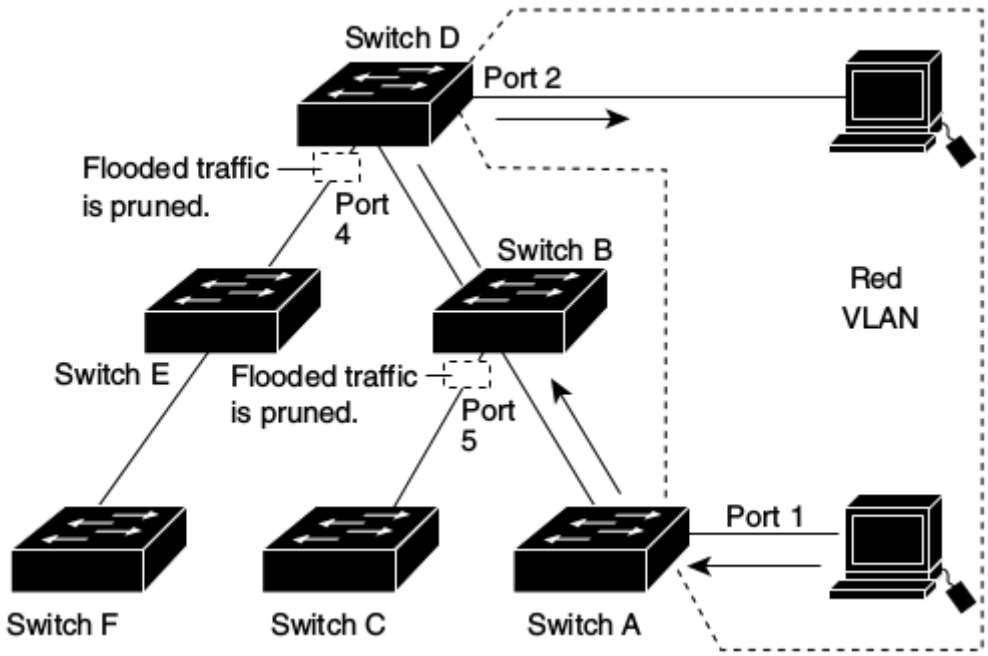
```
(config)# sh vtp password
```

4.4.7. VTP pruning

- Problem:
  - Broadcasts and unknown unicast/multicast frame are flooded everywhere in the broadcast domain included through trunks links
  - Manual editing allowed list is a huge administrative overhead



- Solution: VTP pruning
  - Switches advertise what they need
- All other VLANs are pruned off the trunk link



- Restriction:
  - Pruning does not work in transparent mode. Why?

Pruning eligibility

- When VTP pruning is enabled on a VTP server, pruning is enabled for the entire management domain except for pruning-ineligible VLANS (Vlan 1, 1002-1005, 1006-4094)
- Making VLANs pruning-eligible or pruning-ineligible affects pruning eligibility for those VLANs

on that trunk only (not on all switches in the VTP domain).

- VTP pruning takes effect several seconds after you enable it.

4.4.8. Troubleshooting

http://www.cisco.com/c/en/us/support/docs/lan-switching/vtp/98155-tshoot-vlan.html#topic9

4.5. DTP

4.5.1. Dynamic trunk protocol

- negotiate trunk status
- default to **dynamic auto**
- both sides must be on the same VTP domain or one must be in the NULL domain

Table 4. Trunking configuration options that lead to a working trunk

Configuration Command	Short name	Meaning	To trunk other side must be
switchport mode trunk	Trunk	Always trunks on this end; sends DTP to help other side choose to trunk	On, desirable, auto
switchport mode trunk ; switchport nonegotiate	Nonegotiate	Always trunks on this end; does not send DTP messages (good when other switch is a non-Cisco switch)	On
switchport mode dynamic desirable	Desirable	Sends DTP messages, and trunks if negotiation succeeds	On, desirable, auto
switchport mode dynamic auto	Auto	Replies to DTP messages, and trunks if negotiation succeeds	On, desirable
switchport mode access	Access	Never trunks; sends DTP to help other side reach same conclusion	Never trunks
switchport mode access; switchport nonegotiate	Access (with nonegotiate)	Never trunks; does not send DTP messages	(Never trunks)

Task: Configure an inter-switch link to be in dynamic desirable state

```
(config-if)# switchport mode dynamic desirable
```

Task: Disable DTP for a port administratively configured as a trunk

```
(config-if)# switchport mode trunk
(config-if)# switchport nonegotiate
```

Task: Put the interface into permanent nontrunking mode

```
(config-if)# switchport mode access
```

Task: Display a summary of trunk-related information

```
show interface trunk: Summary of trunk-related information
```

Task: List trunking details for a specified interface

```
show interface <type number> trunk
```

Task: List nontrunking details for a particular interface

```
show interface <type number> switchport
```

Task: Display DTP information for the switch

```
# show dtp
```

Task: Display DTP information for a specific interface

```
# show dtp interface <type slot/number>
```

4.5.2. Trunking between a switch and a Router

Because DTP is not supported on Router

- on the router, create a sub-interface for each desired vlan
- on the switch, disable DTP and manually configure the trunk

Task: Enable trunking but disable DTP for routers

```
! SW1
conf t
int e0/0
  switchport trunk enc dot1q
  switchport mode trunk
  switchport nonegotiate

! R1
conf t
int e0/0.1
  enc dot1q <vlan-id> [native]
```

4.5.3. Verify

What is TOS/TAT in

```
sh dtp interface fa 0/19
```

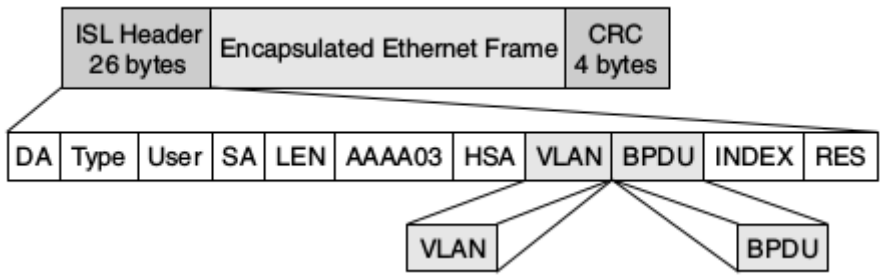
4.6. ISL

4.6.1. Overview

- Inter-Switch Link
- Cisco proprietary
- Provides VLAN trunking
- Supports normal and extended VLANs
- Encapsulates the original header with 26-byte header
- Removes the header at the receiving end

4.6.2. Frame

The ISL frame consists of three fields: the ISL header(26 bytes), the original frame and the FCS (4 bytes)



Field descriptions

DA—Destination Address

- 40-bit
- Multicast address: "0100-0C00-00" or "0300-0C00-00".
- The first 40 bits of the DA field signal the receiver that the packet is in ISL format. ???

TYPE—Frame Type

- 4 bits
- Indicates the type of the original frame
  - 0000: Ethernet
  - 0001: Token Ring
  - 0010: FDDI

- 0011: ATM

USER—User Defined Bits (TYPE Extension)

- 4 bits
- Extends the meaning of the TYPE field
- Default value: "0000"
- For Ethernet frames, the USER field bits "0" and "1" indicate the priority of the packet as it passes through the switch. Whenever traffic can be handled in a manner that allows it to be forwarded more quickly, the packets with this bit set should take advantage of the quick path. It is not required that such paths be provided.
  - XX00 Normal Priority
  - XX01 Priority 1
  - XX10 Priority 2
  - XX11 Highest Priority

SA—Source Address

- 48 bits set to set MAC address of the switch port that transmits the frame.
- May be ignored by the receiving device

LEN—Length

- 16 bits set to the length of the packet in bytes with the exclusion of the DA, TYPE, USER, SA, LEN, and FCS fields.

AAAA03 (SNAP)—Subnetwork Access Protocol (SNAP) and Logical Link Control (LLC)

- 24 bits set to "0xAAAA03".

HSA—High Bits of Source Address

- 24 bits set to 0x00-00-0C (Cisco OUI) of the SA field.

VLAN—Destination Virtual LAN ID

- 15 bits set to the VLAN ID of the frame

BPDU—BPDU and CDP Indicator

- 1 bit set when STP or CDP encapsulates an ISL packet

INDX—Index

- 16 bits set to the port index of the source of the packet as it exits the switch
- Used for diagnostic purposes only
- May be ignored by the receiving bridge

RES—Reserved for Token Ring and FDDI

- 16 bits used when Token Ring or FDDI packets are encapsulated with an ISL frame
  - In the case of Token Ring frames, the Access Control (AC) and Frame Control (FC) fields

are placed here.

- In the case of FDDI, the FC field is placed in the Least Significant Byte (LSB) of this field.

- For Ethernet packets, the RES field should be set to all zeros.

ENCAP FRAME—Encapsulated Frame

- Encapsulated data packet with its own CRC value completely unmodified
- Length from 1 to 24575 bytes

FCS—Frame Check Sequence

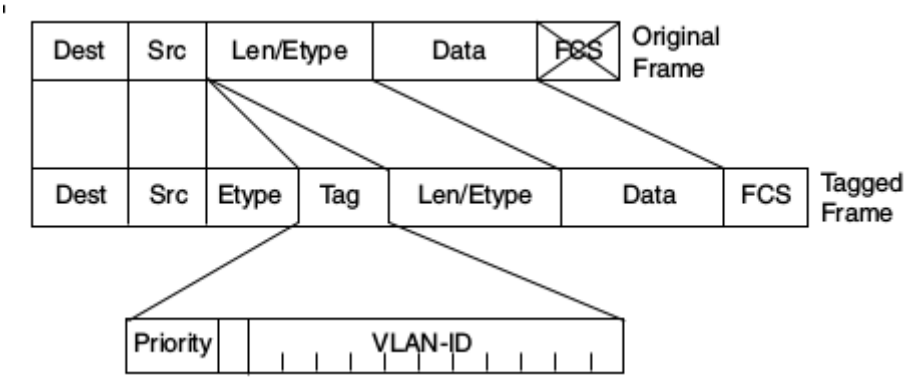
- 4 bytes set by the sending MAC and recalculated by the receiving bridge
- New FCS calculated over the entire ISL packet

4.7. IEEE 802.1Q

4.7.1. Definition

- Tags frames on a trunk
  - Inserts a 4-byte tag into the original frame between the Source Address and the Type/Length field
  - Recomputes the frame check sequence (FCS) before the device sends the frame over the trunk link.
  - Removes the tag at the receiving end
- Does not tag frames on the native VLAN.
  - Must use the same native VLAN on both sides of the trunk
  - Default to VLAN 1
- Supports up to 4096 VLANs
  - Defines a single instance of spanning tree that runs on the native VLAN for all the VLANs in the network.
  - lacks the flexibility and load balancing capability of PVST that is available with ISL.
  - PVST+ offers the capability to retain multiple spanning tree topologies with 802.1Q trunking.

4.7.2. Frame Format



Field descriptions

TPID—Tag Protocol Identifier

- 16 bits
- Value: 08100

Priority

- 3 bits
- Called also user priority or IEEE 802.p
- Indicates the frame priority level
- Can be used to prioritize the traffic

CFI—Canonical Format Indicator

- 1 bit
- Value: 0 if MAC address is in canonical format otherwise 1

VID—VLAN Identifier

- 12 bits
- Identifies the VLAN to which the frame belongs

Ethernet Frame Size with 802.1Q tagging

- Maximum size: 1522 bytes
- Minimum size: 68 bytes

4.7.3. Native VLAN

Task: Configure a native VLAN over a trunk link

```
(config-if)# switchport trunk native vlan <id>
```

4.8. 802.1Q-in-Q tunneling

- Adds a metro tag or PE-VLAN to the 802.1q tagged packets

- Expands the VLAN space by double-tagging frames
- Allows Service Providers
  - to preserve 802.1Q VLAN tags accross WAN links.
  - to provide services such as Internet access on specific VLANs for specific customers, yet providing other services on other VLANs

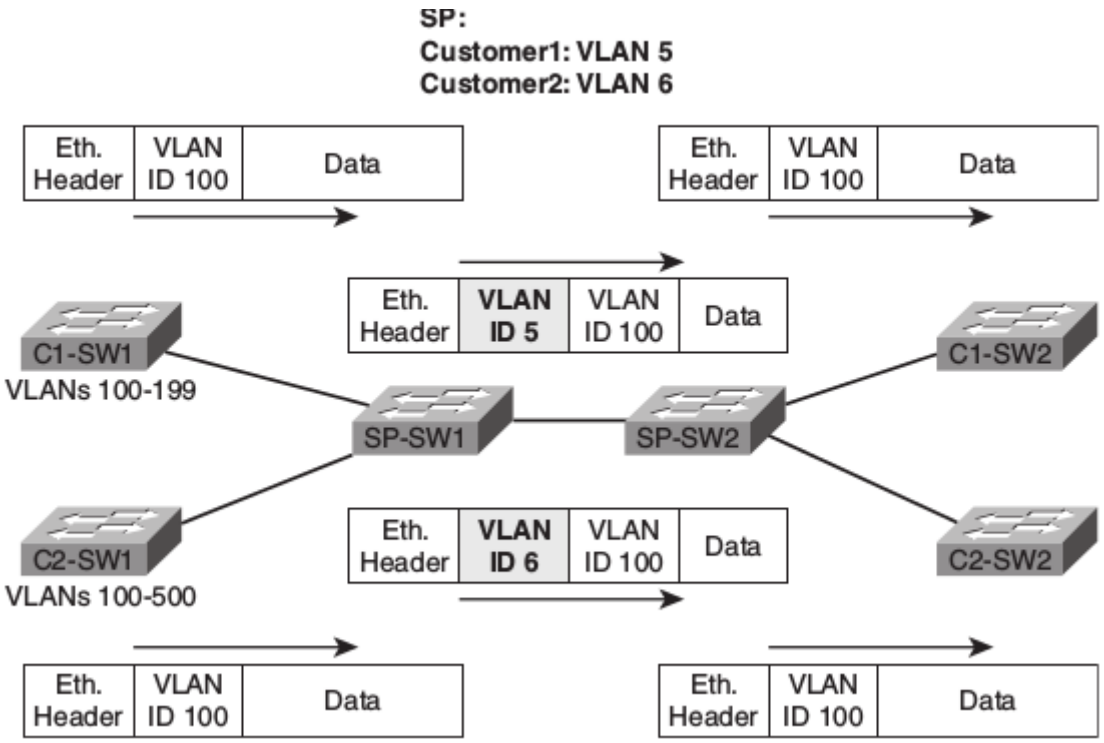
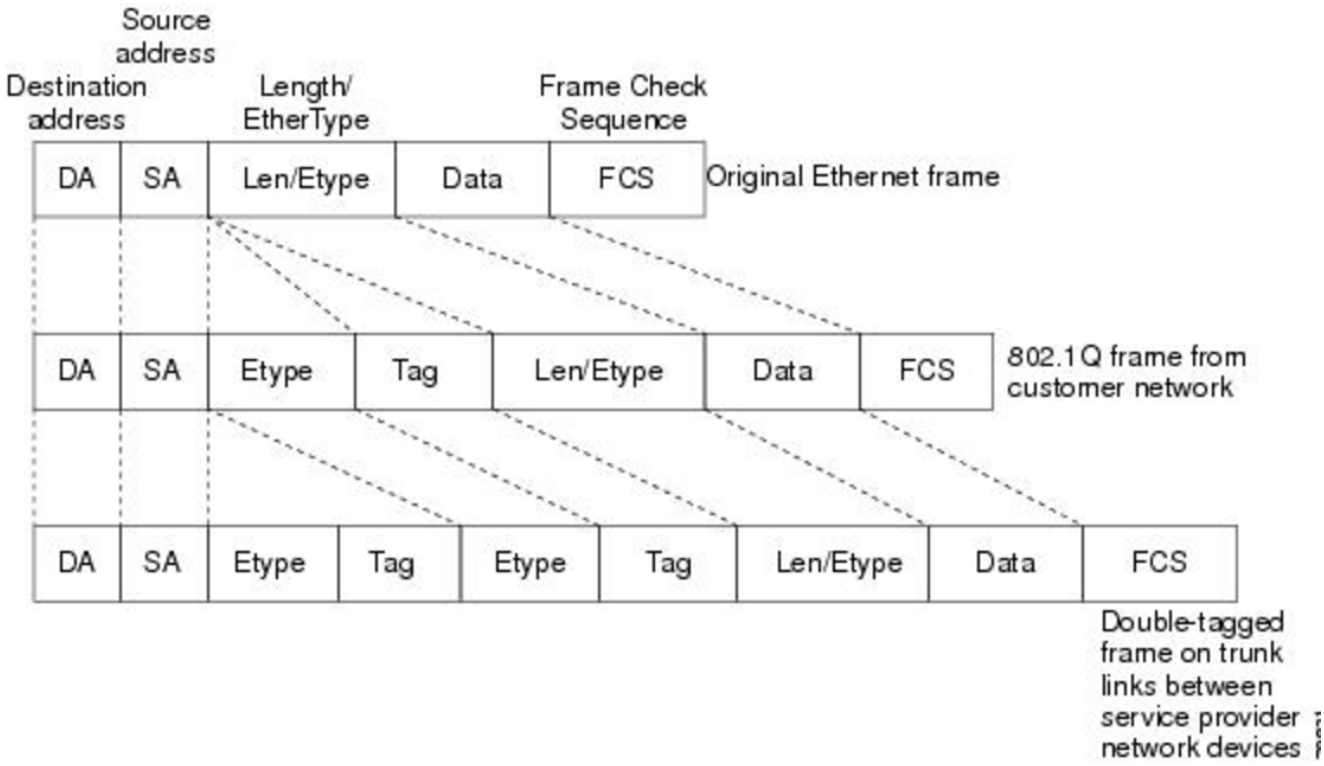


Figure 9. Q-in-Q: basic operation

4.8.1. Frame



Frame size

- Recommended minimum MTU: 1504 bytes
  - default MTU: 1500 bytes
  - outer VLAN tag: 4 bytes

TPID

- Contains the modified tag protocol identifier
- Set to 0x8100 for IEEE 802.1q

The QinQ frame contains the modified tag protocol identifier (TPID) value of VLAN Tags. By default, the VLAN tag uses the TPID field to identify the protocol type of the tag. The value of this field, as defined in IEEE 802.1Q, is 0x8100.

The device determines whether a received frame carries a service provider VLAN tag or a customer VLAN tag by checking the corresponding TPID value. After receiving a frame, the device compares the compares the configured TPID value with the value of the TPID field in the frame. If the two match, the frame carries the corresponding VLAN tag. For example, if a frame carries VLAN tags with the TPID values of 0x9100 and 0x8100, respectively, while the configured TPID value of the service provider VLAN tag is 0x9100 and that of the VLAN tag for a customer network is 0x8200, the device considers that the frame carries only the service provider VLAN tag but not the customer VLAN tag.

In addition, the systems of different vendors might set the TPID of the outer VLAN tag of QinQ frames to different values. For compatibility with these systems, you can modify the TPID value so that the QinQ frames, when sent to the public network, carry the TPID value identical to the value

of a particular vendor to allow interoperability with the devices of that vendor. The TPID in an Ethernet frame has the same position with the protocol type field in a frame without a VLAN tag. In order to avoid problems in packet forwarding and handling in the network, you cannot set the TPID value to any of the values in this table:

Protocol type	Value
ARP	0x0806
PUP	0x0200
RARP	0x8035
IP	0x0800
IPv6	0x86DD
PPPoE	0x8863/0x8864
MPLS	0x8847/0x8848
IS-IS	0x8000
LACP	0x8809
802.1x	0x888E

The QinQ Support feature is generally supported on whatever Cisco IOS features or protocols are supported. For example, if you can run PPPoE on the subinterface, you can configure a double-tagged frame for PPPoE. IPoQinQ supports IP packets that are double-tagged for QinQ VLAN tag termination by forwarding IP traffic with the double-tagged (also known as stacked) 802.1Q headers.

Chapter 5. Spanning tree

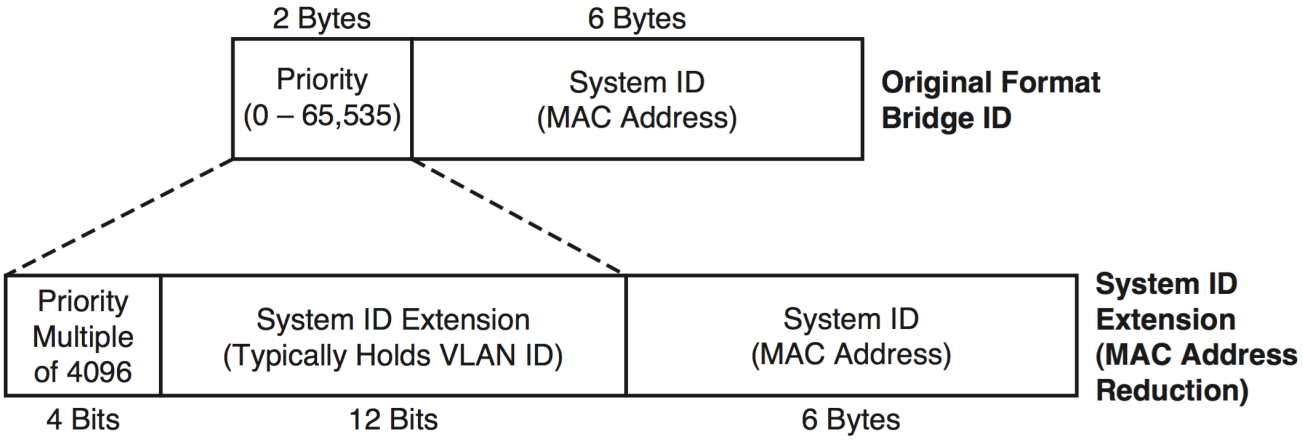
5.1. STP

http://www.CISCO.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_2_se/configuration/guide/3750x_cg/swstp.html

- Creates loop-free layer 2 topology
- Prevents broadcast storms
- STP variations:
  - 802.1d : Common Spanning Tree
  - PVST/PVST+: CISCO per-VLAN Spanning Tree
  - 802.1w: Rapid Spanning Tree Protocol
  - 802.1s: Multiple STP

5.1.1. 802.1d

- Uses BPDU
- Elect one root switch and one designated switch for each segment
- One root port per non-root switch, one designated port for each segment
- Other ports on blocking state
- Steps
  - Elect the root switch with the lowest bridge id (2-byte priority + 6-byte MAC)
  - Determine each switch's root port: with the least cost path to the root
  - Determine the designated port for each segment: the switch that forwards the least cost Hello on the segment
  - If there is a tie, select the lowest port ID
- Original IEEE 802.1d bridge Id
  - 2-byte priority
  - 6-byte MAC adress
- Revised IEEE 802.1d brigde id Priority for MAC address reduction
  - 4 bits : priority multiple of 4096
  - 12 bits : system id extension (vlan id) to support pvst+ and IEEE 802.1s



BPDU

Configuration BPDU

BPDU Field	Length in Octets
Protocol Identifier	2
Protocol Version	1
BPDU Type	1
Flags	1
Root Bridge ID	8
Root Path Cost	4
Sending Bridge ID	8
Sending Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2

Topology Change Notification BPDU

BPDU Field	Length in Octets
Protocol Identifier	2
Protocol Version	1
BPDU Type	1

```
For STP, the Protocol Identifier value is set to 0x0000 and the Protocol
Version is also set to 0x00. The BPDU Type field identifies two kinds of STP
BPDUs: Configuration BPDUs (type 0x00) and Topology Change Notification BPDUs
(type 0x80). The Flags field uses 2 bits out of 8 to handle topology change
events: the Topology Change Acknowledgment flag and the Topology Change flag.
Following the Flags, there is a series of fields identify- ing the root bridge,
distance of the BPDU's sender from the root bridge, the sender bridge's own
identifier, and the identifier of the port on the sender bridge that forwarded
this BPDU. The MessageAge field is an estimation of the BPDU's age since it was
originated by the root bridge. At the root bridge, it is set to 0. Any other
switch will increment this value, usually by 1, before forwarding the BPDU
further. The remaining lifetime of a BPDU after being received by a switch is
MaxAge-MessageAge. Finally, the remaining fields carry the values of STP
timers: MaxAge, HelloTime, ForwardDelay. These timer values always reflect the
timer settings on the root switch. Timers configured on a nonroot switch are
not used and would become effective only if the switch itself became the root switch.
```

- To determine which BPDU out of a pair of configuration BPDUs is superior, they are compared in the following sequence of values, looking for the first occurrence of a lower value: Root Bridge ID, Root Path Cost, Sender Bridge ID, Sender Port ID, Receiver Port ID (not included in BPDU; evaluated locally)
- Each port in STP stores/remembers the superior BPDU it has either sent (DP port) or received (RP and Blocking Ports). Essentially, each port stores the DP's BPDU—whether it is the port itself that is Designated or it is a neighbor's port. Should a port store a received BPDU, it must be received again within a time interval of MaxAge-MessageAge seconds; otherwise it will expire after this period. This expiry is always driven by the timers in the BPDU, that is, according to timers of the root switch.

Root Bridge

- Election with hello BPDU

```
Each switch begins its STP logic by creating and sending an Hello BPDU message,
claiming itself to be the root switch.
If a switch hears a superior Hello to its own Hello bridge ID,
it stops claiming to be root by ceasing to originate and send Hellos.
Instead, the switch starts forwarding the superior Hellos received from the
superior candidate.
Eventually, all switches except the switch with the lowest bridge ID cease to
originate Hellos;
that one switch wins the election and becomes the root switch.
```

Task: Force election of a root bridge

```
# spanning-tree vlan <id> root
```

Root port

- RP is upstream facing towards Root bridge
- Lowest root path cost (cumulative cost of all links to get to the root)
  - cost = advertised cost in the BPDU hello + cost on the receiving post
- Cost based on inverse bandwith

Table 5. Default port costs

Speed	original	revised	802.1D-2004
10 Mbps	100	100	2000000
100 Mbps	10	19	200000
1 Gbps	1	4	20000
10 Gbps	1	2	2000

Task: Choose default STP path cost (original or revised)

```
(config)# spanning-tree pathcost method {short | long}
```

Tie breaker when a switch receives multiple Hellos with equal cost

1. Lowest bridge id
2. Lowest port priority
3. Lowest port number

Designated port

- Designated switch: send the Hello with the lowest advertised cost for the segment
- DP: port that forward frames onto that segment
- DP are downstream facing away from root bridge
- Elected based on lowest root path cost, BID, port ID

Blocking ports

- Receive BPDUs
- Discard all other traffic
- Cannot send traffic
- Do not send Hellos

Convergence

- Steady operations: one Root bridge, one RP on each non-root bridge, one DP on each segment, blocking state

1. Root switch generates a Hello every 2 seconds
2. Each RP on non-root switch receives a copy of the root's Hello
3. Each DP updates and forwards the Hello out
4. Each blocking port receives a copy of the Hello from the DP without forwarding it

Topology change notification

more at understand new topology changes

1. A switch experiencing the STP port state change sends a TCN BPDU out its Root Port; it repeats this message every Hello time until it is acknowledged.
2. The next switch receiving that TCN BPDU sends back an acknowledgment via its next forwarded Hello BPDU by marking the Topology Change Acknowledgment (TCA) bit in the Hello.
3. The switch that was the DP on the segment in the first two steps repeats the first two steps, sending a TCN BPDU out its Root Port, and awaiting acknowledgment from the DP on that segment.

By each successive switch repeating Steps 1 and 2, eventually the root receives a TCN BPDU. Once received, the root sets the TC flag on the next several Hellos, which are forwarded to all switches in the network, notifying them that a change has occurred. A switch receiving a Hello BPDU with the TC flag set uses the short (Forward Delay time) timer to time out entries in the CAM.

Table 6. Transitioning from blocking to forwarding

State	Forward data frames	Learn source MAC	Stable?
Blocking	No	No	Yes
Listening	No	No	No
Learning	No	Yes	No
Forwarding	Yes	Yes	Yes
Disabled	No	No	Yes

Timers

Hello timer

- 2 seconds
- Interval at which the root sends Hellos

- Forward delay

- 15 seconds
- Time that switch leaves a port in listening state and learning state
- also used for the short CAM timeout timer

- Maxage

- 20 seconds
- Time without hearing a Hello before believing that the root has failed

PVST+

- Per-VLAN STP : for better load balancing
  - One instance of legacy STP per VLAN
  - CISCO ISL support
- PVST+
  - One instance of legacy STP per VLAN
  - CISCO ISL and 802.1q support
  - Interoperability between CST and PVST
- default mode on most Catalyst platforms
- allows root bridge/port placement per VLAN
- Non-CISCO + 802.1q ⇒ one Common Spanning Tree over vlan 1
- When mixing CISCO and non CISCO switches with 802.1q trunking,
  - Send bpdu to multicast destination MAC of 0100.0CCC.CCCD

Task: Display Spanning-Tree information

```
# sh spanning-tree root
# sh spanning-tree vlan 1 root detail
```

Optimizing, improving spanning tree

PortFast

- Used on access ports connected to end users devices not other switches
- Puts the port into forwarding state immediately
- Prevent them to generate TCNs
- Can generate loops if another switch is connected. so must be used with bpdu guard and root guard features

Task: Enable portfast

```
(config-if)# spanning-tree portfast
(config)# spanning-tree portfast default
```

UplinkFast

- Used on access layer switches that have multiple uplinks to distribution/core switches

- Immediately replaces a lost RP with an alternate RP
- Increases the root and all port priority so the switch does not become root or transit switch
- Time-out the correct entries in their CAMs but doesnt use the TCN process. Instead, finds all the MAC addresses of local devices and sends one multicast frame with each local addresses as the source MAC causing all the other switches to update their CAMs. The access switch also clears out the rest of the entries in its own CAM.

Task: Enable uplink fast

```
(config)#  spanning-tree uplinkfast [max-update-rate rate]
```

BackboneFast

- Used in core switches to detect indirect link failures to the Root
- Do not wait for Maxage to expire when another switch's direct link fails
- Send a Root Link Query out the port in which the missing Hello should arrive. The RLQ asks the neighboring switch if that neighboring switch is still receiving Hellos from the root. IF that neighbor had a direct link failure, it can tell the original switch via another RLQ that this path to the root is lost. Once known, the switch experiencing the indirect link failure can go ahead and converge without waiting for Axage to expire
- All switches must have backbone fast configured

```
(config)#  spanning-tree backbonefast
```

bpdu filter

- Filter BPDUs in and out

bpdu guard

- Puts a portfast enabled port into the errdisable state when a BPDU is received and shuts down the port
- The port must be manually re-enabled or it can be recovered automatically through the errdiable timeout function.
- A port configured with bpdu guard will not be put into the root-inconsistent state.

loop guard

- Prevents non-designated ports from inadvertently forming layer 2 switching loops if the flow of bpdus is interrupted.
- Puts the port into the loop-inconsistent state when the steady flow of BPDUs is interrupted
- Only used on point-to-point links
- Can be used with **UDLD aggressive mode** to get extra protection.

root guard

- Prevent a port from becoming a root port when receiving a superior bpdu (e.g. inferior priority + mac)
- It is enabled on ports other than the root port and on switches other than the root.
- Puts the port in **root-inconsistent** state (no data flow) until it stops receiving superior BPDUs. No traffic is forwarded.
- Enforce the root bridge placement by ensuring the the port on which root guard is enabled is the designated port.

http://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/10588-74.html

- Enforce the root bridge placement
- Ensures that the port on which root guard is enabled is the designated port.

Loop Guard

When normal BPDUs are no longer received, the port does not go through normal STP convergence, but rather falls into an STP loop-inconsistent state.

In all cases, the formerly blocking port that would now cause a loop is prevented from migrating to a forwarding state. With both types of UDLD, the switch can be configured to automatically transition out of err-disabled state. With Loop Guard, the switch automatically puts the port back into its former STP state when the original Hellos are received again.

5.1.2. 802.1w Rapid STP

- Improves convergence by
  - Waiting for only 3 missed Hellos on an RP before flushing the CAM instead of 10 with 802.1d
  - Bypass listening state
  - Includes natively CISCO PortFast, UplinkFast, BackboneFast
  - Add backup DP when multiple ports connected to the same segment
- Backward compatible with 802.1d
- All bridges generate BPDUs every Hello interval
- Use a single BPDU
  - No TCN BPDU
  - Protocol version = 0x02
  - The Flags field has been updated. In 802.1D STP BPDUs, only 2 bits out of 8 are used: TC (Topology Change) and TCA (Topology Change Acknowledgment). RSTP uses the 6 remaining bits as well to encode additional information: Proposal bit, Port Role bits, Learning bit, Forwarding bit, and Agreement bit. The TCA bit is not used by RSTP. This change allows implementing the Proposal/Agreement mechanism and also allows a BPDU to carry information about the originating port's role and state, forming the basis of RSTP's Dispute mechanism, protecting against issues caused by unidirectional links.

RSTP link types

- **Point-to-point**: Switch to Switch (default if full-duplex port)
- **Shared** : Switch to hub (default if half-duplex port)
- **Edge**: Switch to single end-user device

Task: Set the RSTP link-type

```
spanning-tree link-type { point-to-point | shared }
```

RSTP port types

- **Edge** :
- **Non-Edge**: default

RSTP port states

- Default to discarding at start

Administrative state	802.1d	802.1w
Disabled	Disabled	Discarding
Enabled	Blocking	Discarding
Enabled	Listening	Discarding
Enabled	Learning	Learning
Enabled	Forwarding	Forwarding

RSTP port roles

Root Port

- Same role as 802.1d RP

Designated Port

- Same role as 802.1d DP
- Default role at boot

Alternate Port

- An alternate root port
- Same concept as CISCO UplinkFast feature
- Protects against the loss of a switch's RP by keeping track of the AP with a path to the root

Backup Port

- No equivalent CISCO feature
- Protects against losing the DP attached to a shared liknk when the switch has another

physical port attached to the same shared segment

NOTE	root bridge ports are all designated port unless 2 or more ports of the root bridge are connected together.

NOTE	a port needs to receive BPDUs to stay blocked.

Task: Configure Rapid PVST

```
(config)#  spanning-tree mode rapid-pvst
```

NOTE	• Rapid PVST+ immediately deletes dynamically learned MAC address entries when it receives a topology change instead of a timer used by PVST+ or MST

Proposal/Agreement

TODO See 5th Edition

Topology Change Handling

TODO See 5th Edition

5.1.3. 802.1s Multiple Spanning Trees

- Multiple VLANs mapped to the same STP instance.
- Enable load balancing
- Improves fault tolerance of the network because a failure in one instance or forwarding path does not affect other instances.
- Uses 802.1w for rapid convergence
- Highly scalable
  - Switches with same instance, configuration revision number and name form a **region**
  - Different regions see each other as virtual bridges
- Each switch have three attributes:
  - Alphanumeric configuration name (32 bytes)
  - Configuration number (2 bytes)
  - 4096-element table that associates each of the potential 4096 VLANs to a map ???

MST Region

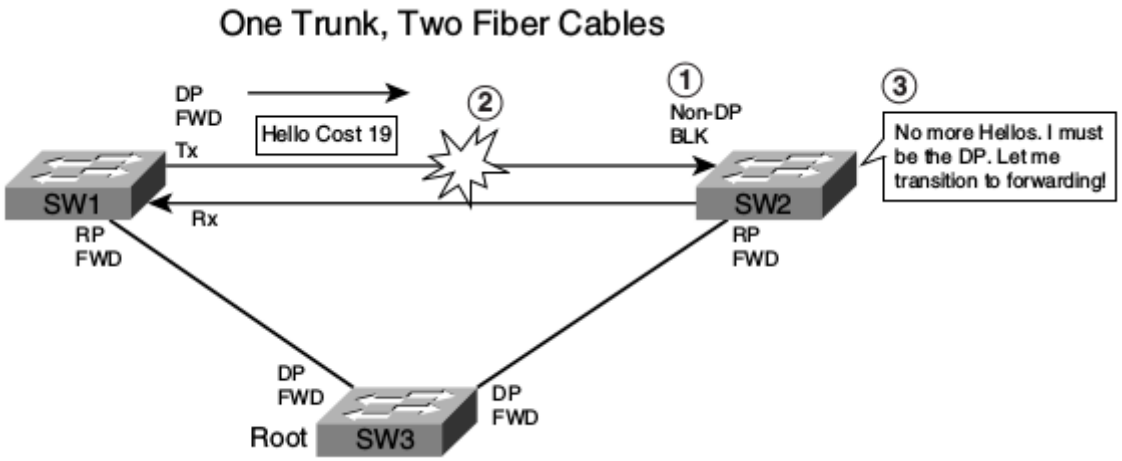
MST Revision Number

MST Instance

5.1.4. Protecting Against Unidirectional Link Issues

UDLD

- Unidirectional links:
  - One of the 2 transmission path has failed but not both
  - Due to miscabling, cutting on fiber cable, unplugging one fiber, GBIC problems, …
  - Can cause a loop as the previously blocking port will move to a forwarding state



- Solutions:

UDLD **u***ni***d***irectional* **l***ink* **d***etection*

Uses Layer 2 messaging to decide when a switch can no longer receive frames from a neighbor. The switch whose transmit interface did not fail is placed into an err-disabled state.

UDLD aggressive mode

Attempts to reconnect with the other switch (eight times) after realizing no messages have been received. If the other switch does not reply to the repeated additional messages, both sides become err-disabled.

Bridge Assurance

```
The Bridge Assurance, applicable only with RPVST+ and MST and only on
point-to-point links, is a further extension of the idea used by Loop Guard.
Bridge Assurance modi- fies the rules for sending BPDUs. With Bridge Assurance
activated on a port, this port always sends BPDUs each Hello interval, whether
it is Root, Designated, Alternate, or Backup. BPDUs thus essentially become a
Hello mechanism between pairs of intercon- nected switches. A Bridge
Assurance–protected port is absolutely required to receive BPDUs. If no BPDUs
are received, the port will be but into a BA-inconsistent block- ing state
until it starts receiving BPDUs again. Apart from unidirectional links, Bridge
Assurance also protects against loops caused by malfunctioning switches that
completely stop participating in RPVST+/MST (entirely ceasing to process and
send BPDUs) while opening all their ports. At the time of this writing, Bridge
Assurance was supported on selected Catalyst 6500 and Nexus 7000 platforms.
Configuring it on Catalyst 6500 Series requires activating it both globally
using spanning-tree bridge assurance and on ports on STP point-to-point link
types toward other switches using the spanning-tree portfast network interface
command. The neighboring device must also be configured for Bridge Assurance.
```

Dispute Mechanism

```
The Dispute mechanism is yet another and standardized means to detect a
unidirectional link. Its functionality is based on the information encoded in
the Flags field of RST and MST BPDUs, namely, the role and state of the port
forwarding the BPDU. The principle of operation is very simple: If a port
receives an inferior BPDU from a port that claims to be Designated Learning or
Forwarding, it will itself move to the Discarding state. Cisco has also
implemented the Dispute mechanism into its RPVST+. The Dispute mechanism is not
available with legacy STP/PVST+, as these STP versions do not encode the port
role and state into BPDUs. The Dispute mechanism is an integral part of
RSTP/MST and requires no configuration.
```

storm control

unicast flooding

5.1.5. Troubleshooting

flapping port that is generating BPDUs with the TCN bit set

5.2. MST

5.2.1. operations

cst → 1 stp for all vlan pvst → 1 stp for each vlan mst → 1 stp per instance

5.2.2. readings

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_2_se/configuration/guide/3750x_cg/swmstp.html

Chapter 6. EtherChannel

6.1. EtherChannel

- EtherChannel aggregates bandwidth of up to 8 physical links
- Consists of two parts:
  - Port-channel interface: logical interface representing the bundle
  - Member interfaces: physical links part of the bundle
- can be any type of interface: Layer 2 access, trunk, tunnel or layer 3 routed
- Configured as either Layer 2 or Layer 3 interfaces.
- To be part of a PortChannel, both sides must agree on:
  - Same speed and duplex settings
  - If not trunking, same access VLAN
  - If trunking, same trunk type, allowed VLANs, and native VLAN
  - On a single switch, each port in a PortChannel must have the same STP cost per VLAN on all links in the PortChannel
  - No ports with SPAN configured
- When several EtherChannel bundles exist between two switches, STP blocks one of the bundles to prevent redundant links. When spanning tree blocks one of the redundant links, it blocks one EtherChannel, thus blocking all the ports belonging to this EtherChannel link.
- Where there is only one EtherChannel link, all physical links in the EtherChannel are active because STP sees only one (logical) link.
- If a link within an EtherChannel fails, traffic previously carried over that failed link changes to the remaining links within the EtherChannel. A trap is sent for a failure, identifying the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link of the EtherChannel.
- Each EtherChannel has a logical port-channel interface numbered from 1 to 64. The channel groups are also numbered from 1 to 64.
- When a port joins an EtherChannel, the physical interface for that port is shut down.
- When the port leaves the port-channel, its physical interface is brought up, and it has the same configuration as it had before joining the EtherChannel.

6.1.1. Link aggregation protocol

- PAgP
  - Maximum 8 ports
- LACP
  - Maximum 16 ports

- Maximum 8 active ports and 8 standby ports

Task: Verify which negotiation protocol has been used for the EtherChannel

```
# show etherchannel protocol
```

Task: Specify the link aggregation protocol globally

```
(config-if)# channel-protocol {pagp | lacp}
```

- The **channel-group** interface configuration command can also set the mode for the EtherChannel
- If you set the protocol by using **channel-protocol**, the setting is not overriden by the **channel-group** interface configuration command.

6.1.2. Layer 2 EtherChannels

- Logical interfaces are dynamically created when using **channel-group** command.

Task: Configure layer 2 EtherChannels

```
conf t
interface <type slot/number>
  switchport mode {access | trunk}
  channel-group n mode {active | passive | on | {auto [non-silent] | desirable [non-
silent]  }
```

6.1.3. Layer 3 EtherChannels

Task: Create the port channel logical interface

```
conf t
interface port-channel <number>
  no switchport
  ip address <a.b.c.d> <mask>
```

Task: Assign the physical interfaces to the layer 3 port channel

```
conf t
interface <type id>
  no switchport
  no ip address
  channel-group n mode {active | passive | on | {auto [non-silent] | desirable [non-
silent]  }
```

- Always issue the **no switchport** command before the **channel-group** command
- If L3 port-channel configured properly, the **show etherchannel summary** command should show **RU** for routed and in use

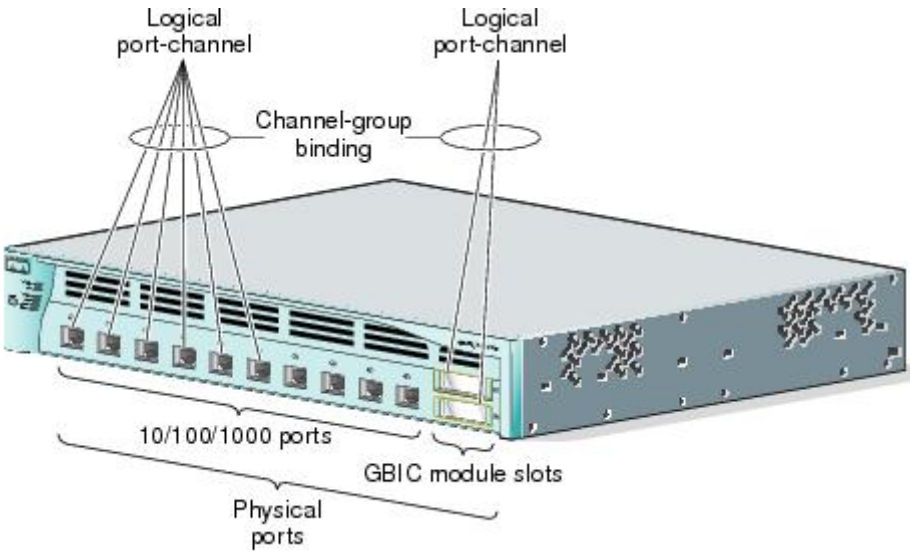


Figure 10. Relationship of Physical Ports, Logical Port Channels, and Channel Groups

6.1.4. EtherChannel modes

Table 7. EtherChannel modes

Cisco PAgP	802.1AD LACP	Description
on	on	disable negotiation and forces the port into the portChannel
off	off	disable negotiation and prevents the ports to be part of the portChannel
desirable	active	initiates the negotiation
auto	passive	waits on other side to start negotiation

Task: Display EtherChannel status

```
# show etherchannel [group-number]
```

PAgP and LACP Interaction with Other Features

- DTP and CDP send and receive packets over the physical interfaces in the EtherChannel.
- PAgP and LACP transmit PDUs on the lowest numbered VLAN on the interfaces enable for (desirable,auto or active,passive)
- STP sends packets over the first interface in the Etherchannel
- The MAC address of a Layer 3 EtherChannel is the MAC address of the first interface in the port-channel.

Load balancing and forwarding modes

- Load balancing between member interface based on a combination of
  - Source MAC address
  - Destination MAC address
  - Source IP address
  - Destination IP address
- Uses only source MAC address by default

Task: Configure the EtherChannel load-balancing method

```
(config)# port-channel load-balance { dst-ip | dst-mac | src-dst-ip | src-dst-mac |
src-ip | src-mac}
```

Task: Display the EtherChannel load-balancing method

```
# show etherchannel load-balance

EtherChannel Load-Balancing Configuration:
src-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source MAC address
IPv4: Source MAC address
IPv6: Source MAC address
```

6.1.5. EtherChannel Misconfiguration guard

- This mechanism makes an assumption that if multiple ports are correctly bundled into a Port-channel at the neighbor side, all BPDUs received over links in this Port-channel must have the same source MAC address in their Ethernet header, as the Port-channel interface inherits the MAC address of one of its physical member ports. If BPDUs sourced from different MAC addresses are received on a Port-channel interface, it is an indication that the neighbor is still treating the links as individual, and the entire Port-channel will be err-disabled
- Enabled by default

Task: Deactivate EtherChannal misconfig guard

```
(config)# no spanning-tree etherchan- nel guard misconfig
```

6.2. LACP

6.2.1. Overview

- IEEE 802.3ad
- Automatic creation of port channels
- Multicast address IEEE 802.3 Slow Protocols: 0180-C200-0002
- EtherType value: 0x8809
- Timers: hellos every second during hand shake
- Maximum: 16 ports with max 8 active

Restrictions

Modes

Passive

- Does not initiate LACP negotiation but responds to LACP packets
- Default mode

Active

- Initiate LACP negotiation by sending LACP packets

On

- Forces the interface to the channel without PAgP or LACP

Working Etherchannel for On-On, Passive-Active, Active-Active

6.2.2. LACP hot-standby ports

- Only 8 LACP links can be active at one time
- Any additional links are in hot-standby mode
- If one of the active links becomes inactive, a hot-standby link becomes active in its place
- Each link is assigned a unique priority in this order
  - LACP system priority (1..65535, default: 32768)
  - System ID (the switch MAC address)
  - LACP port priority
  - Port number
- In priority comparisons, lower values have higher priority.
- To determine which ports are active and which ports are hot standby,
  - Select the master switch with a low system priority and system-id
  - Select the master ports with the low port priority and number. The port-priority and port-number of the slave switch are not used.

Task: Check which ports are in the hot-standby mode

```
# show etherchannel summary
```

Task: Configure the LACP system priority

```
(config)# lacp system-priority <priority>
```

Task: Show the LACP system priority

```
# show lacp sys-id
```

Task: LACP port priority

```
(config-if)# lacp port-prioriy
```

6.2.3. LACP Port-channel MaxBundle feature

- Control the number of ports allowed to be bundled into the etherchannel
- Allows hot-standby ports with fewer bundled ports

Task: Configure the maximum number of bundled ports allowed in a LACP port channel

```
(config-if)# lacp max-bundle
```

6.2.4. LACP Port-Channel Min-links feature

- Only for LACP Etherchannel
- Prevents low-bandwidth interface from becoming active
- Causes LACP etherChannels to become inactive if they have too feww active members ports to supply the required minimum bandwith

Task: Configure the minimum number of member ports that must be in the link-up state and bundled in the etherchannel for the port channel interface to transition to the link-up state

```
(config-if)# port-channel min-link n
```

6.3. PAgP

6.3.1. Overview

- **P**ort **Ag**gregation **P**rotocol
- Cisco proprietary

- Automatic creation of a EtherChannel.
- Sends PagP packets every 30 seconds to multicast 0100-0CCC-CCCC
- Same destination address than CDP, UDLD, VTP, and DTP.
- Checks for configuration consistency and manages link additions and failures between two switches.
- Protocol value: 0x104
- Cannot be enabled on cross-stack EtherChannel

Task: Display PAgP status

```
# show pagp [channel-group-number]
```

6.3.2. Modes

Auto

- Never initiates PAgP communications but instead listen passively for any received PAgP packets before creating an EtherChannel with the neighboring switch.
- Default mode

Desirable

- Initiates negotiations with other interfaces by sending PAgP packets.

On

- Forces the interface to channel without PAgP.
- Do not exchange PAgP packets.

Etherchannel formed for on-on, desirable-auto, desirable-desirable combinations.

6.3.3. Physical vs Aggregate learners

Switches running PAgP are classified as:

PAgP physical learners

- learn MAC addresses using the physical ports within the EtherChannel instead of via the logical EtherChannel link.
- forward traffic to addresses based on the physical port via which the address was learned. The switch will send packets to the neighboring switch using the same port in the EtherChannel from which it learned the source address.

Aggregate learners

- learns addresses based on the aggregate or logical EtherChannel port.
- transmit packets to the source by using any of the interfaces in the EtherChannel.
- Aggregate learning is the default.

By default, PAgP is not able to detect whether a neighboring switch is a physical learner. Therefore, when configuring PAgP EtherChannels on switches that support only physical learning, the learning method must be manually set to physical learning. It is important when running in this mode, to set the load-distribution method to source-based distribution so that any given source MAC address is always sent on the same physical port.

Task: Configure the PAgP learning method

```
(config-if)# pagp learn-method {physical-port | aggregation-port>
```

Task: Verify the PAgP learning method

```
# show pagp [channel-group-number] internal
```

6.3.4. Priority

- Range: 0..255
- Default: 128
- The higher the priority, the more likely that the port will be used for PAgP transmission

Task: Assign a priority so that the selected port is chosen for packet transmission.

```
(config-if)# pagp port-priority <priority>
```

6.3.5. Restrictions

While PAgP allows for all links within the EtherChannel to be used to forward and receive user traffic, there are some restrictions:

- DTP and CDP send and receive packets over all the physical interfaces in the EtherChannel, while PAgP sends and receives PAgP PDU only from interfaces that are up and have PAgP enabled for auto or desirable modes.
- When an EtherChannel bundle is configured as a trunk port, the trunk sends and receives PAgP frames on the lowest numbered VLAN. STP always chooses the first operational port in an EtherChannel bundle.
- When configuring additional STP features such as Loop Guard on an EtherChannel, remember that if Loop Guard blocks the first port, no BPDUs will be sent over the channel, even if other ports in the channel bundle are operational. This is because PAgP will enforce uniform Loop Guard configuration on all of the ports that are part of the EtherChannel group.

6.3.6. Configuration

Validate the port that will be used by STP to send packets and receive packets

```
Switch#show pagp neighbor
Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
A - Device is in Auto mode.        P - Device learns on physical port.

Channel group 4 neighbors
Partner     Partner             Partner         Partner   Group
Port        Name                Device ID       Port       Age  Flags   Cap.
Gi1/1/3     Switch.1            00c5.a003.0080   Gi0/1     4s   SC      10001
Gi1/1/4     Switch.1            00c5.a003.0080   Gi0/2     3s   SC      10001
```

STP will send packets only out of port Gi1/1/3 because it is the first operational interface. If that port fails, STP will send packets out of Gi1/1/4.

6.3.7. Silent mode

Task: Configure a switch port for nonsilent operation

```

```

Task: Configure a switch port for nonsilent operation

```

```

TODO You can also configure a single interface within the group for all transmissions and use other interfaces for hot standby. The unused interfaces in the group can be swapped into operation in just a few seconds if the selected single interface loses hardware-signal detection.

Chapter 7. SPAN , RSPAN and ERSPAN

- SPAN (Switch Port Analyzer) mirrors monitored (TX,RX or Both) traffic on source ports or VLANs to a destination port for analysis.

7.1. Local SPAN sessions

- Source and destination on the same switch image::span-topology.png[SPAN Topology]

Task: Display SPAN status

```
Switch# show monitor session

Session 1
==========

Type              : Local Session
Source Ports      :
    RX Only       : None
    TX Only       : None
    Both          : Fa0/4
Source VLANs      :
    RX Only       : None
    TX Only       : None
    Both          : None
Source RSPAN VLAN : None
Destination Ports : Fa0/5
    Encapsulation: DOT1Q
          Ingress: Enabled, default VLAN = 5
Reflector Port    : None
Filter VLANs      : None
Dest RSPAN VLAN   : None
```

7.1.1. Source

- Can not mix source ports and source VLANs in a single session
- Monitored traffic directions can be
  - Rx : before any modification or processing by ACL or QoS or VACL
  - Tx : after all modification and processing performed by the switch.
  - Both: by default

Task: Configure span source ports/VLANs

```
monitor session <1-66> source {interface <id> | vlan <id>} [, | -] [both | rx | tx]
```

- (Optional) [, | -] Specify a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
- A single session can include multiple sources (ports or VLANs), defined in a series of commands, but you cannot combine source ports and source VLANs in one session.

Source ports

- Can be physical interfaces
- Can be port-channel logical interfaces with port-channel numbers in (1..48)
- Can be an access port, trunk port, routed port, or voice VLAN port.
- Cannot be a destination port

Source VLANs

- All active ports in the source VLAN are included as source ports and can be monitored in either or both directions.
- On a given port, only traffic on the monitored VLAN is sent to the destination port.
- If a destination port belongs to a source VLAN, it is excluded from the source list and is not monitored.
- If ports are added to or removed from the source VLANs, the traffic on the source VLAN received by those ports is added to or removed from the sources being monitored.
- You cannot use filter VLANs in the same session with VLAN sources.
- You can monitor only Ethernet VLANs.
- Ignores CDP, BPDU, VTP,, DTP and PAgP frames unless **encapsulation replicate** is configured

7.1.2. Destination port

- Must be a physical port
- Cannot be a source port
- By default, send packets untagged
  - can replicate the source interface encapsulation
- By default, disable the ingress traffic forwarding
  - can accept incoming packets with dot1q, isl or untagged
- Only one SPAN/RSPAN session can send traffic to a single destination port, cannot be used by two SPAN sessions
- Only monitored traffic passes through the SPAN destination port
- Entering SPAN configuration commands does not remove previously configured SPAN parameters. Enter the **no monitor session {session_number | all | local | remote}** global configuration command to delete configured SPAN parameters.

- For local SPAN, outgoing packets through the SPAN destination port carry the original encapsulation headers—untagged, ISL, or IEEE 802.1Q If the encapsulation replicate keywords are specified. If the keywords are not specified, the packets are sent in native form. For RSPAN destination ports, outgoing packets are not tagged.
- You can configure a disabled port to be a source or destination port, but the SPAN function does not start until the destination port and at least one source port or source VLAN are enabled.
- You cannot mix source VLANs and filter VLANs within a single SPAN session.
- Up to 64 SPAN destination ports can be configured on a switch

Task: Configure the destination port for a SPAN session

```
(config)# monitor <session-number> destination interface <interface-id>
                                    [encapsulation replicate]
                                    [ingress {dot1q vlan <vlan-id> | isl | untagged
vlan <vlan-id>} ]
```

7.1.3. VLAN Filtering

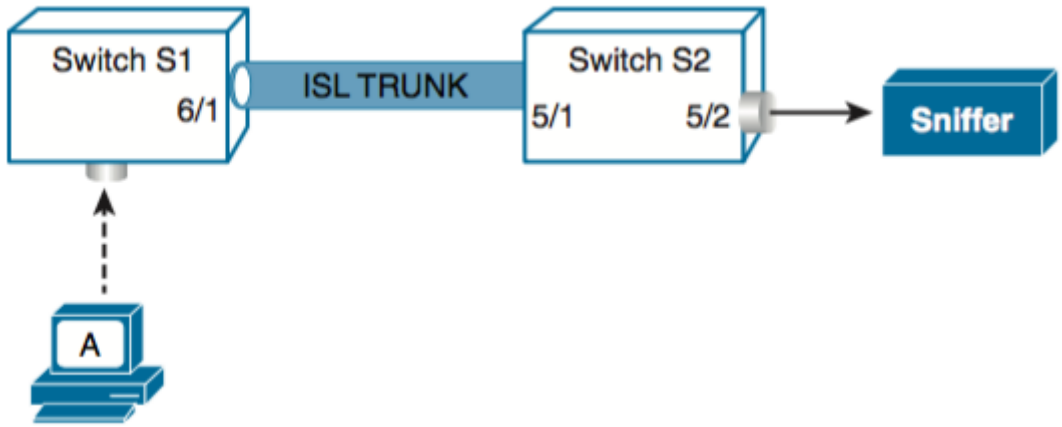
- To limit SPAN traffic monitoring on trunk source ports to specific VLANs by using VLAN filtering.
- Applies only to trunk ports or to voice VLAN ports.
- Applies only to port-based sessions
- Not allowed in sessions with VLAN sources.
- When a VLAN filter list is specified, only those VLANs in the list are monitored on trunk ports or on voice VLAN access ports.
- SPAN traffic coming from other port types is not affected by VLAN filtering; that is, all VLANs are allowed on other ports.
- VLAN filtering affects only traffic forwarded to the destination SPAN port and does not affect the switching of normal traffic.

Task: Limit SPAN source to specific VLANs

```
(config)# monitor <session-number> filter vlan <vlan-ids>
```

7.2. Remote SPAN sessions

RSPAN consists of at least one RSPAN source session, an RSPAN VLAN, and at least one RSPAN destination session.



Restrictions and considerations

When RSPAN is enabled, each packet being monitored is transmitted twice, once as normal traffic and once as a monitored packet. Therefore monitoring a large number of ports or VLANs could potentially generate large amounts of network traffic.

7.2.1. RSPAN VLAN

- Can be propagated to all switches by VTP if RSPAN VLAN < 1005
- Must be created manually on extended-range VLAN
- Can not be vlan 1, 1002-1005
- Can served multiple RSPAN source/destination sessions

Restrictions

- You can apply an output ACL to RSPAN traffic to selectively filter or monitor specific packets. Specify these ACLs on the RSPAN VLAN in the RSPAN source switches.
- For RSPAN configuration, you can distribute the source ports and the destination ports across multiple switches in your network.
- RSPAN does not support BPDU packet monitoring or other Layer 2 switch protocols.
- The RSPAN VLAN is configured only on trunk ports and not on access ports. To avoid unwanted traffic in RSPAN VLANs, make sure that the VLAN remote-span feature is supported in all the participating switches.
- Access ports (including voice VLAN ports) on the RSPAN VLAN are put in the inactive state.
- RSPAN VLANs are included as sources for port-based RSPAN sessions when source trunk ports have active RSPAN VLANs. RSPAN VLANs can also be sources in SPAN sessions. However, since the switch does not monitor spanned traffic, it does not support egress spanning of packets on any RSPAN VLAN identified as the destination of an RSPAN source session on the switch.

Task: Configure RSPAN VLAN on all participating switches

```
(config)# vlan <rspan-vlan-id>
(config-vlan)# remote-span
```

7.2.2. RSPAN source session

- Must be configured on the monitored port's switch

Task: Configure the RSPAN source session

```
monitor session <session-number> source {interface interface-id | vlan vlan-id} [, |
-] [both | rx | tx]
monitor session session_number destination remote vlan <rspan-vlan-id>
```

7.2.3. RSPAN destination session

- Takes all packets received on the RSPAN VLAN, strips off the VLAN tagging, and presents them on the destination port.
- Excludes Layer 2 control

Task: Configure the RSPAN destination session on a different switch (not the switch on which the source session was configured)

```
(config)# monitor session <session-number> source remote vlan <rspan-vlan-id>
(config)#  monitor session <session-number> destination interface <interface-id>
```

7.3. Interaction with other features

Routing

- SPAN does not monitor routed traffic.
- RSPAN only monitors traffic that enters or exits the switch, not traffic that is routed between VLANs.

STP

- A destination port does not participate in STP while its SPAN or RSPAN session is active.
- The destination port can participate in STP after the SPAN or RSPAN session is disabled.
- On a source port, SPAN does not affect the STP status. STP can be active on trunk ports carrying an RSPAN VLAN.

CDP

- A SPAN destination port does not participate in CDP while the SPAN session is active.
- After the SPAN session is disabled, the port again participates in CDP.

VTP

- You can use VTP to prune an RSPAN VLAN between switches.

VLAN and trunking

- You can modify VLAN membership or trunk settings for source or destination ports at any time.

- However, changes in VLAN membership or trunk settings for a destination port do not take effect until you remove the SPAN destination configuration.
- Changes in VLAN membership or trunk settings for a source port immediately take effect, and the respective SPAN sessions automatically adjust accordingly.

EtherChannel

- You can configure an EtherChannel group as a source port but not as a SPAN destination port.
- When a group is configured as a SPAN source, the entire group is monitored.
- If a physical port is added to a monitored EtherChannel group, the new port is added to the SPAN source port list.
- If a port is removed from a monitored EtherChannel group, it is automatically removed from the source port list.
- A physical port that belongs to an EtherChannel group can be configured as a SPAN source port and still be a part of the EtherChannel.
- In this case, data from the physical port is monitored as it participates in the EtherChannel. However, if a physical port that belongs to an EtherChannel group is configured as a SPAN destination, it is removed from the group. After the port is removed from the SPAN session, it rejoins the EtherChannel group. Ports removed from an EtherChannel group remain members of the group, but they are in the inactive or suspended state.
- If a physical port that belongs to an EtherChannel group is a destination port and the EtherChannel group is a source, the port is removed from the EtherChannel group and from the list of monitored ports.

Multicasting

- Multicast traffic can be monitored.
- For egress and ingress port monitoring, only a single unedited packet is sent to the SPAN destination port.
- It does not reflect the number of times the multicast packet is sent.

Private VLAN

- A private-VLAN port cannot be a SPAN destination port.

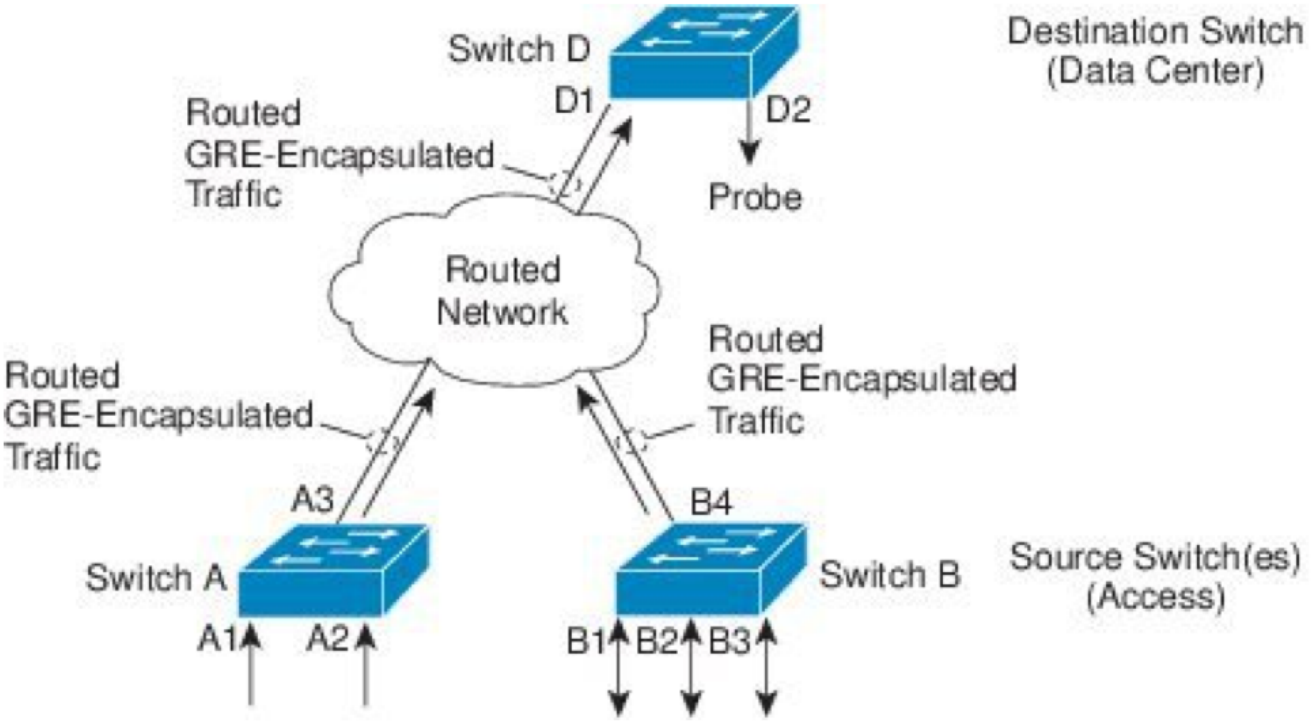
Secure port

- A secure port cannot be a SPAN destination port.
- For SPAN sessions, do not enable port security on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable port security on any ports with monitored egress.
- An IEEE 802.1x port can be a SPAN source port. You can enable IEEE 802.1x on a port that is a SPAN destination port; however, IEEE 802.1x is disabled until the port is removed as a SPAN destination.
- For SPAN sessions, do not enable IEEE 802.1x on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable IEEE

802.1x on any ports that are egress monitored.

7.4. Encapsulated RSPAN

- ERSPAN consists of an ERSPAN source session, routable ERSPAN GRE encapsulated traffic, and an ERSPAN destination session.
- Supported only on high-end switch



7.4.1. ERSPAN source session

Task: Configure ERSPAN source session

```
(config)# monitor session <id> type erspan-source
(config-mon-erspan-src)# source { interface <interface-id> | vlan <vlan-ids>
[rx|tx|both]}
(config-mon-erspan-src)# destination
(config-mon-erspan-src-dst)# erspan-id <erspan-flow-id>
(config-mon-erspan-src-dst)# mtu <size>
(config-mon-erspan-src-dst)# origin ip address <a.b.c.d> [force]
(config-mon-erspan-src-dst)# no shutdown
```

7.4.2. ERSPAN destination session

Task: Configure ERSPAN destination session

```
(config)# monitor session <id> type erspan-destination
(config-mon-erspan-dst)# destination interface <interface-id>
(config-mon-erspan-dst)# source
(config-mon-erspan-dst-src)# erspan-id <erspan-flow-id>
(config-mon-erspan-dst-src)# mtu <size>
(config-mon-erspan-dst-src)# ip address <a.b.c.d> [force]
(config-mon-erspan-dst-src)# no shutdown
```

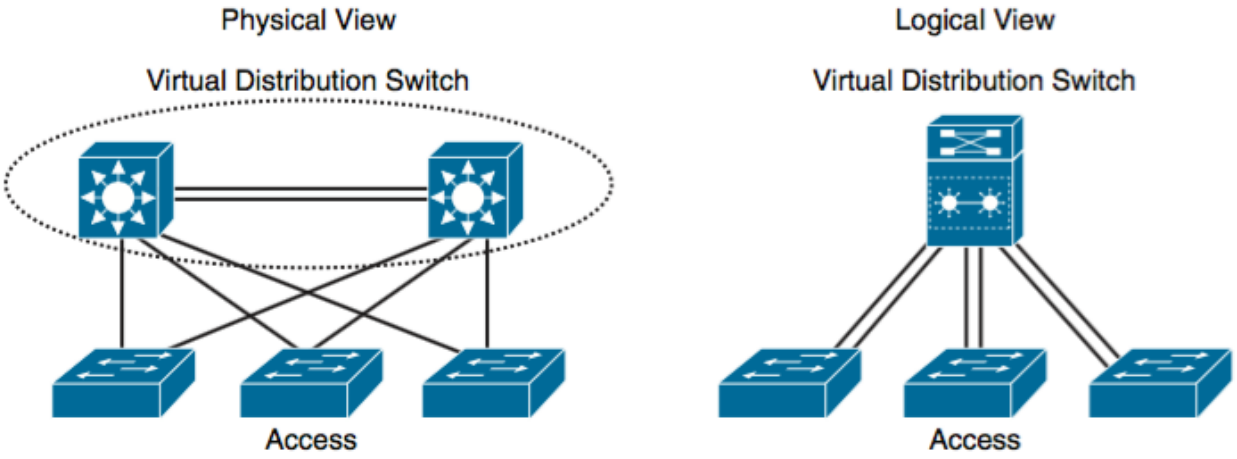
7.4.3. ESPAN dummy MAC address rewrite

- Supports customizaed MAC value for WAN interface and tunnel interface
- Monitor the traffic going through WAN interface

Task: Configure ESPAN dummy MAC address

```
(config)# monitor session <session-id> type erspan-source
(config-mon-erspan-src-dst)# s-mac <mac-address>
(config-mon-erspan-src-dst)# d-mac <mac-address>
```

Chapter 8. Virtual Switch System



- VSS makes two physical switches to act and appear as one single logical network element.
- VSS manages the redundant links from access switches as single Multi-chassis Etherchannel
  - No need for spanning-tree to block one of the links
  - two active links instead of one 1/10/40b interfaces

8.1. VSS Active and Standby Switch

- Uses VLSP to negotiate the active and standby roles at start

Task: Configure VSS domain number and switch number

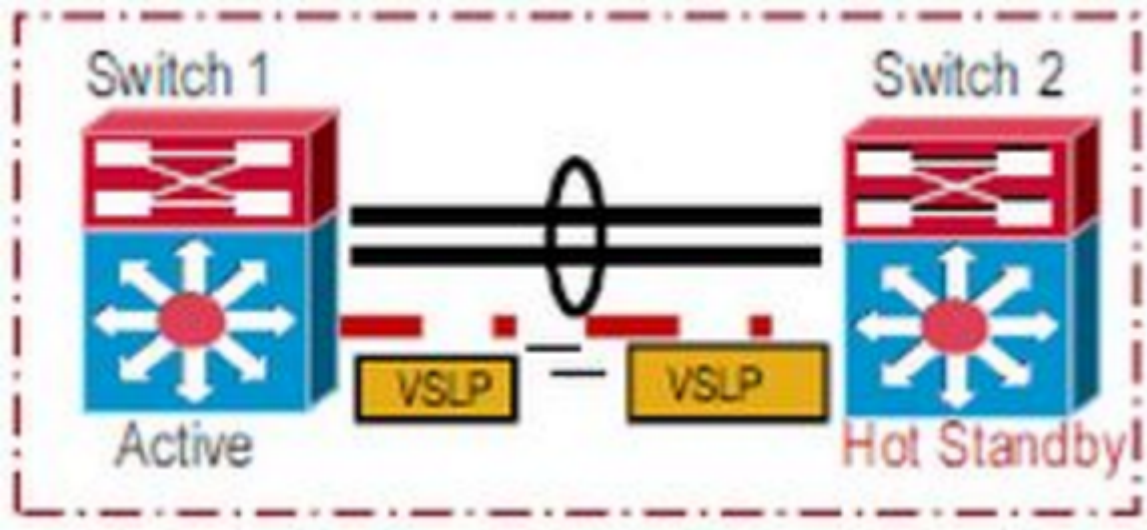
```
(config)# switch virtual domain <1..255>
(config-vs-domain)# switch [1 | 2]
```

Task: Configure VSS switch priority

```
(config-vs-domain)# switch [1 | 2 ] priority [<number>]
```

NOTE 1 lowest priority 255 highest priority 100 default

8.2. Virtual Switch Link



- Normally built as etherchannel with up to 8 links
- carries system control information (hot-standby supervisor programming, line card status, Distributed Forwarding Card (DFC) card programming, system management, diagnostics, …)
- carries user data traffic when necessary

Task: Create VSL

```
(config)#interface port channel 5
(config-if)# switchport
(config-if)# switch virtual link 1
(config-if)# no shut
(config-if)# exit
#! add physical interface to port channel
(config)# interface range gi 7/4 - 5
(config-if)# channel group 5 mode one
(config-if)# exit
```

Task: Convert the switch to virtual

```
(config)# switch convert mode virtual
```

Task: Displays the VSS information

```
# sh switch virtual [role | link]
```

Chapter 9. WAN

9.1. HDLC

- High-Level Data Link Control
- Layer 2 on point-to-point links
- developed by the ISO (ISO 3309)
- modified by Cisco by adding a proprietary 2-byte Type field to the frame.
  - enabled by default on IOS serial links
  - On a Cisco router, HDLC encapsulation can only connect with another Cisco router

9.1.1. HDLC frame format

9.1.2. Encapsulation

- default

Task: Set encapsulation to hdlc

```
(config-if)# encapsulation hdlc
```

Task: Display Statistics

```
sh controllers serial
```

9.1.3. Clock Rate

- Automatically set

Task: Modify the clock rate

```
(config-if)# clock-rate <bps>
```

Task: Specify the clock rate is in the network

```
(config-if)# clock-rate line
```

9.2. PPP

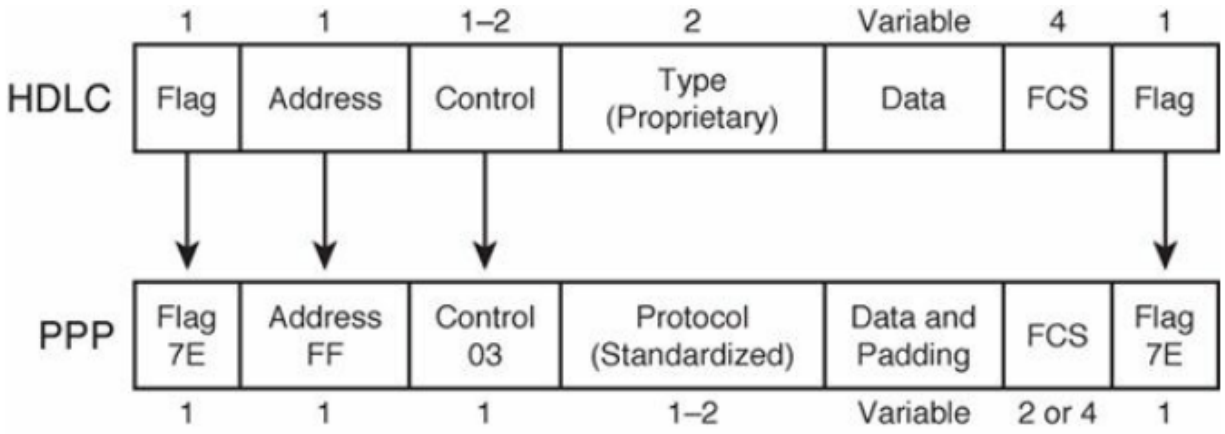
Cisco Cloud Routers Configuration guides | WAN | PPP

- error detection, error recovery, standard protocol Type field, supports synchronous and asynchronous links

- RFC 1661

9.2.1. PPP Frame Format

- Replaces proprietary type with standard protocol
- Adds padding so the frame has even number of bytes



9.2.2. PPP LCP

- link control protocol
- controls features independent of any Layer 3 protocol

Task: Configure PPP

```
(config-if)# encapsulation ppp
```

LCP operations:

```
When a PPP serial link first comes up—for example, when a router senses the
Clear to Send (CTS), Data Send Read (DSR), and Data Carrier Detect (DCD) leads
come up at the physical layer—LCP begins parameter negotiation with the other
end of the link. For example, LCP controls the negotiation of which
authentication methods to attempt, and in what order, and then allows the
authentication protocol (for example, CHAP)
to complete its work. After all LCP negotiation has completed
successfully, LCP is considered to be “up.”
```

LCP features

- LQM link quality monitoring: drop if % of error frames above a configured value
- looped link detection: drop link if a router receives its own randomly chosen magic number
- layer 2 load balancing: fragment frames over multilink PPP
- authentication: chap, pap

configuration

- minimal with **encapsulation ppp**
- optional authentication, quality

```
(config-if)# encapsulation ppp
(config-if)# ppp quality <percent>
(config-if)# ppp authentication {chap | pap}
```

Task: Drop the link if router receives its own magic number

```
```

LQM

- When LQM is enabled, every keepalive period is sent to Link Quality Reports (LQRs) in place of keepalives. All incoming keepalives are responded to properly.
- If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.
- LQM is incompatible with Multilink PPP

Task: Monitor PPP link Quality

```
(config-if)# ppp quality <percent>
```

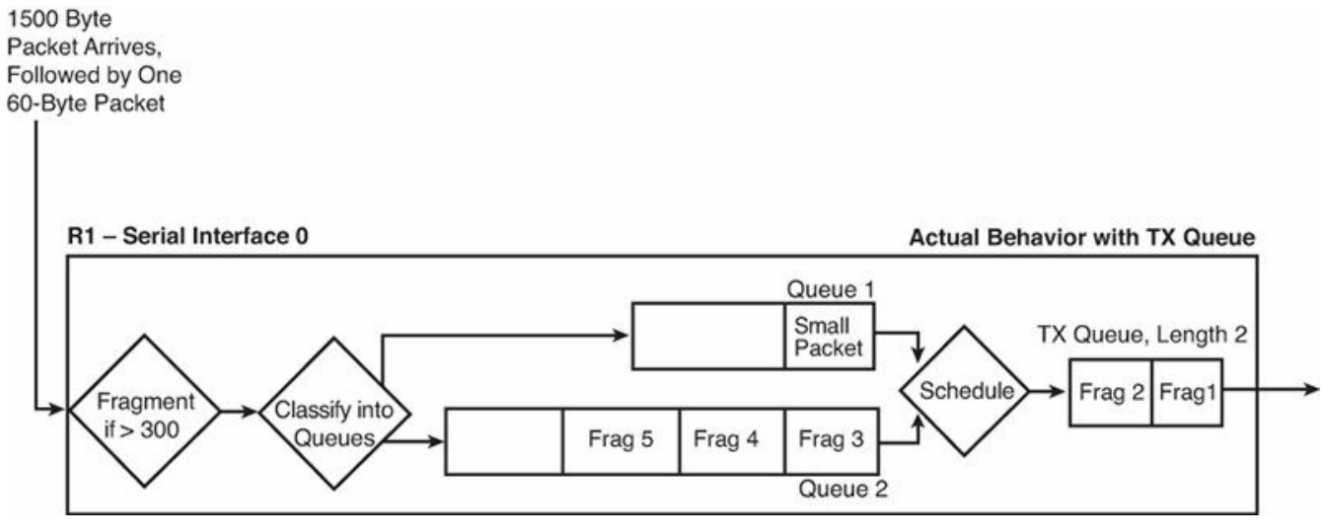
9.2.3. multilink PPP

- originally intended to combine multiple ISDN B-channels without requiring any Layer 3 load balancing
- now load balance traffic accross any type of point-to-point serial link
- add a header (2 or 4 bytes) to allow reassembly on the receiving end
- configuration with multilink interfaces or virtual templates
- fragmenting each data link layer frame, either based on the number of parallel links or on a configured fragmentation delay.
  - sends the fragments over different links at the same time.
  - adds a header (4 or 2 bytes for Sequence Number and Flags bit) to allow reassembly on the receiving end, MLP adds a header (either 4 or 2 bytes)

LFI

- LFI (link fragmentation and interleaving)
- prevents small, delay sensitive packets from having to wait on longer, delay-insensitive packets to be completely serialized out an interface.
- the queuing scheduler generally LLQ on the multilink interface determines the next packet to

send:



Task: Allow interleave

```
(config-if)# ppp multilink interleave
```

Task: Define LFI fragment size

```
(config-if)# ppp multilink fragment-delay <microseconds>
```

defines the fragment size based on "Size" = x * "Bandwidth"

Example

```
interface Multilink1
  bandwidth 256
  ip address 10.1.34.3 255.255.255.0
  encapsulation ppp
  ppp multilink
  ppp multilink group 1
  ppp multilink fragment-delay 10
  ppp multilink8 interleave
  service-policy output queue-on-dsc
```

9.2.4. PPP compression

- uses L2 payload compression (ip + tcp + data + DL) : best with longer packet
- TCP header compression (ip + tcp)
- RTP header compression (ip + udp + rtp)
- payload compression works best with longer packets, and header with shorter packets
- header compression : achieves better compression ration 10:1 to 20:1

layer 2 compression

- options: LZS (Lempel-Ziv Stacker), MPPC (microsoft point-to-point compression), Predictor
- LZS use more CPU and less RAM than Predictor algorithm and have better compression ratio
- stacker: supports hdlc, ppp, FR, ATM
- mppc: ppp, atm
- predictor: ppp, atm
- configuration with a matching **compress** command under each interface on both end of the links
- once configured, ppp starts ccp (compression control protocol) which is another NCP

header compression

- configured with legacy commands or MQC commands
- legacy under the serial (ppp) or multilink interface
- **ip tcp header-compression [passive]**
- **ip rtp header-compression [passive]**
- add also MQC commands

9.2.5. PPP Authentication

Task: Enable PPP authentication

```
ppp authentication {chap | chap pap | pap chap | pap} [if-needed] [<list-name> |
default] [callin]
```

Task: debug ppp authentication

```
debug ppp authentication
```

read understanding debug ppp negotiation

9.3. PPPoE

- used for digital subscriber line (DSL) Internet access because the public telephone network uses ATM for its transport protocol; therefore, Ethernet frames must be encapsulated in a protocol supported over both Ethernet and ATM.
- The PPP Client feature permits a Cisco IOS router, rather than an endpoint host, to serve as the client in a network. This permits multiple hosts to connect over a single PPPoE connection.
- In a DSL environment, PPP interface IP addresses are derived from an upstream DHCP server using IP Configuration Protocol (IPCP). Therefore, IP address negotiation must be enabled on the router's dialer interface. This is done using the ip address negotiated command in the dialer interface configuration.

- Because of the 8-byte PPP header, the MTU for PPPoE is usually set to 1492 bytes so that the entire encapsulated frame fits within the 1500-byte Ethernet frame. A maximum transmission unit (MTU) mismatch prevents a PPPoE connection from coming up. Checking the MTU setting is a good first step when troubleshooting PPPoE connections.
- Because PPPoE introduces an 8-byte overhead (2 bytes for the PPP header and 6 bytes for PPPoE), the MTU for PPPoE is usually decreased to 1492 bytes so that the entire encapsulated frame fits within the 1500-byte Ethernet frame. Additionally, for TCP sessions, the negotiated Maximum Segment Size is clamped down to 1452 bytes, allowing for 40 bytes in TCP and IP headers and 8 bytes in the PPPoE, totaling 1500 bytes that must fit into an ordinary Ethernet frame. A maximum transmission unit (MTU) mismatch can prevent a PPPoE connection from coming up or from properly carrying large datagrams. Checking the MTU setting is a good first step when troubleshooting PPPoE connections.

9.3.1. PPPoE server

- on the ISP side

TODO: Replace the task below with step-by-step instructions

Task: Create a Broad Band Aggregation group

```
(config)# bba-group pppoe <name>
(config-bba-group)# virtual-template 1
```

Task: Limit the number of sessions on the associated MAC

```
(config-bba-group)# sessions per-mac limit <number>
```

Task: Create the virtual template interface

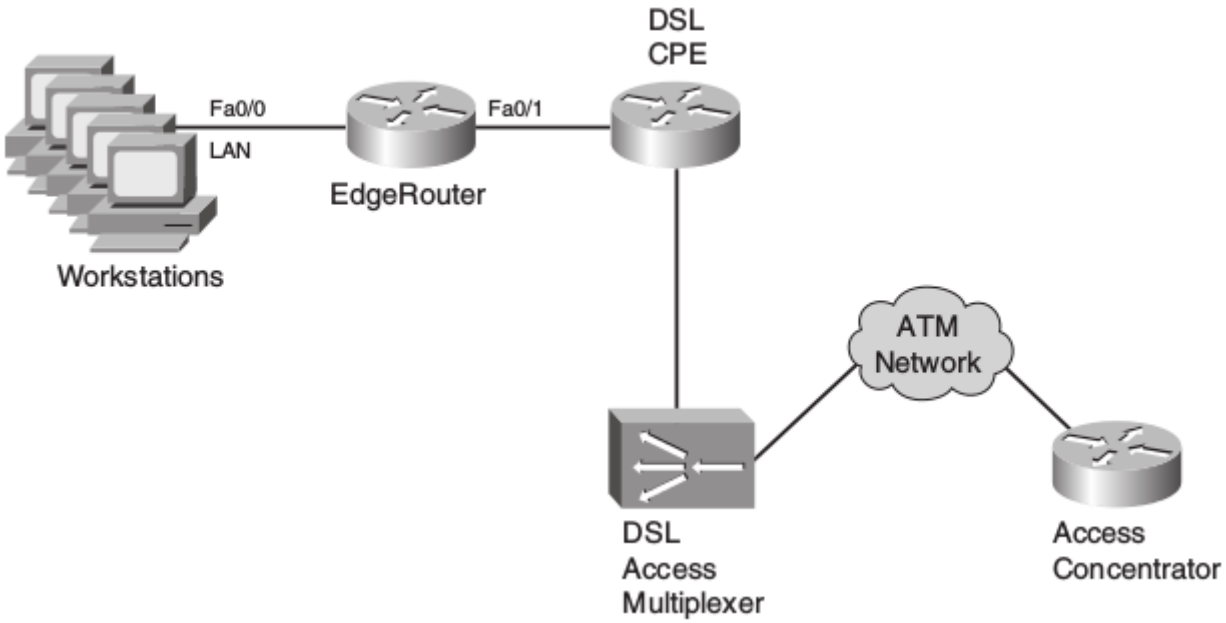
```
(config)# interface virtual-template 1
(config-if)# ip address 10.0.0.1 255.255.255.0
(config-if)# peer default ip address pool <pool-name>

(config)# ip local pool <pool-name> <ip-start> <ip-finish>
```

Task: Enable PPPoE group on the interface

```
(config-if)#
(config)# interface f0/0
(config-if)# no ip address
(config-if)# pppoe enable group MyGroup
(config-if)# no shutdown
```

9.3.2. PPPOE client



Example of config on the Edge router

```
# conf t
(config)# interface fa0/1
(config-if)# ip address 192.168.100.1 255.255.255.0
(config-if)# ip nat inside
(config)# interface fa0/1
(config-if)# pppoe-client dial-pool-number 1
(config-if)# exit
(config)# interface dialer1
(config-if)# mtu 1492
(config-if)# encapsulation ppp
(config-if)# ip address negotiated
(config-if)# ppp authentication chap

!The remaining CHAP commands have been omitted for brevity.

(config-if)# ip nat outside
(config-if)# dialer pool 1
(config-if)# dialer-group 1
(config-if)# exit
(config)# dialer-list 1 protocol ip permit
(config)# ip nat inside source list 1 interface dialier1 overload
(config)# access-list 1 permit 192.168.100.0 0.0.0.255
(config)# ip route 0.0.0.0 0.0.0.0 dialer1
```

Task: Verify PPPoE connectivity

```
show pppoe session
```

Task: Debug

```
debug pppoe [data | errors | events | packets]
```

9.3.3. PPPoE authentication

TODO: Add section from configuration guides

9.4. Ethernet WAN

EWAN

- Virtual Private LAN Services (VPLS),
- Multi-Protocol Label Switching (MPLS),
- Any-Transport Over MPLS (ATOM),
- Dot1Q-in-Dot1Q Tunnels (QnQ Tunnels),
- Metro-Ethernet.

9.4.1. VPLS

- Virtual Private LAN Service
- various WAN connections (over either IP or MPLS networks)
- uses QoS for audio and video
- provide multipoint Ethernet LAN services, or Transparent LAN Service (TLS).
  - A multipoint network service is one that allows a customer edge (CE) endpoint or node to communicate directly with all other CE nodes associated with the multipoint service.
  - By contrast, using a point-to-point network service such as ATM, the end customer typically designates one CE node to be the hub to which all spoke sites are connected. In this scenario, if a spoke site needs to communicate with another spoke site, it must communicate through the hub, and this requirement can introduce transmission delay.
- To provide multipoint Ethernet capability, the IETF VPLS drafts describe the concept of linking virtual Ethernet bridges using MPLS Pseudo-Wires (PW). As a VPLS forwards Ethernet frames at Layer 2, the operation of VPLS is exactly the same as that found within IEEE 802.1 bridges in that VPLS will self-learn the source MAC address to port associations, and frames are forwarded based upon the destination MAC address. If the destination address is unknown, or is a broadcast or multicast address, the frame is flooded to all ports associated with the virtual bridge.
- Although the forwarding operation of VPLS is relatively simple, the VPLS architecture needs to be able to perform other operational functions, such as
  - Autodiscover other provider edges (PE) associated with a particular VPLS instance
  - Signaling of PWs to interconnect VPLS virtual switch instances (VSI)

Loop avoidance

- MAC address withdrawal

9.4.2. Metro-Ethernet

- Ethernet on the metropolitan-area network (MAN) can be used as pure Ethernet, Ethernet over MPLS, or Ethernet over Dark Fiber, but regardless of the transport medium, we have to recognize that in network deployments requiring medium-distance backhaul or metropolitan (in the same city) connectivity, this Ethernet WAN technology is king. Why do we have so many different types of Metro-E solutions? The answer is that each has advantages and disadvantages. As an example, pure Ethernet-based deployments are cheaper but less reliable and scalable, and are usually limited to small-scale or experimental deployments. Dark Fiber–based deployments are useful when there is an existing infrastructure already in place, whereas solutions that are MPLS based are costly but highly reliable and scalable, and as such are used typically by large corporations.
- MPLS-based Metro-Ethernet network uses MPLS in the service provider's network. The subscriber will get an Ethernet interface on copper (for example, 100BASE-TX) or fiber (such as 100BASE-FX). The customer's Ethernet packet is transported over MPLS, and the service provider network uses Ethernet again as the underlying technology to transport MPLS. So MPLS-based Metro-E is effectively Ethernet over MPLS over Ethernet.
- Label Distribution Protocol (LDP) signaling can be used to provide site-to-site signaling for the inner label (VC label) and Resource Reservation Protocol-Traffic Engineering (RSVP-TE), or LDP can be used to provide the network signaling for the outer label.
- It should also be noted that a typical Metro-Ethernet system has a star network or mesh network topology, with individual routers or servers interconnected through cable or fiber-optic media. This is important when it becomes necessary to troubleshoot Metro-Ethernet solutions.

-

Part II : Layer 3 Technologies

Chapter 10. IP addressing

10.1. IPv4

- RFC 791
- IP Ethernet protocol: 0x0800

10.1.1. IP packet format

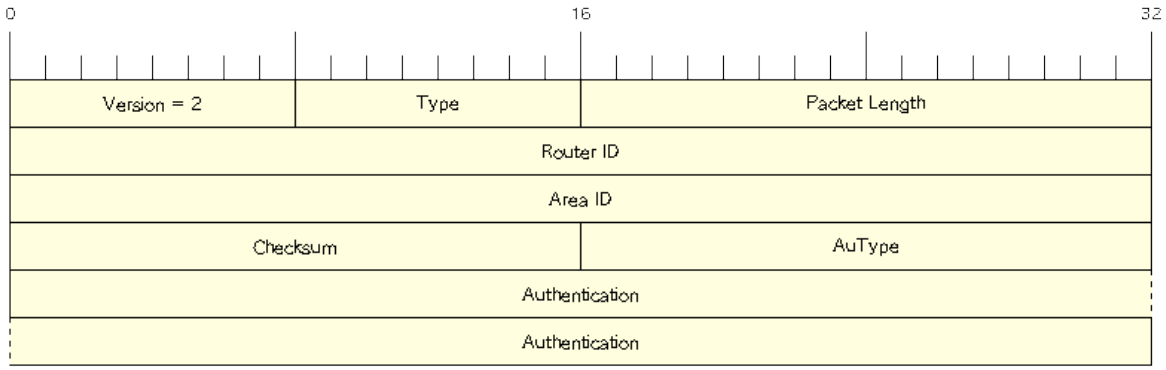


Figure 11. IP header format

IP Header Length (IHL)

Datagram header length in 32-bit words.

Type-of-Service

Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance. Currently referred to as Differentiated Services Code Point (DSCP) (6 bits).

Total Length

Specifies the length, in bytes, of the entire IP packet, including the data and header.

Identification

Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

Flags

Consists of a 3-bit field of which the two low-order (least-significant) bits control fragmentation. The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

Fragment Offset

Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

Time-to-Live

Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

Protocol

Indicates which upper-layer protocol receives incoming packets after IP processing is complete.

Header Checksum

Helps ensure IP header integrity.

Options

Allows IP to support various options, such as security.

Data

Contains upper-layer information.

10.1.2. IP address

- 32-bits written in "dotted decimal"
- Classes: A,B,C,D,E
- Classless : prefix + host

Task: Assign an IP address to an interface

```
(config-if)# ip address <a.b.c.d> <e.f.g.h> [secondary]
```

Task: Display the IP parameters for the interface

```
show ip interface
```

Task: Display the IP networks the device is connected to

```
show ip route connected
```

10.1.3. CIDR

- Classless interdomain routing
- Defined in RFS 1517-1520
- Administrative assigment of large address blocks and the related summarized routes for the purpose of reducing the size of the Internet routing table
- Enable by default

Task: Enable classless addressing

```
(config)# ip classless
```

Task: Specify the format in which netmask appear for the current session

```
(config)# line vty <first> <last>
(config-line)# term ip netmask-format {bitcount | decimal | hexadecimal}
```

Task: Specify the format in which netmask appear for the current line

```
(config)# line vty <first> <last>
(config-line)# term ip netmask-format {bitcount | decimal | hexadecimal}
```

10.1.4. Private addressing

- RFC 1918
- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

10.1.5. VLSM

- Variable length subnet mask

10.1.6. Subnet zero

Task: Allow IP subnet zero

```
(config)# ip subnet-zero
```

10.1.7. Unnumbered interfaces

- Borrow the IP address of another interface
- Only point-to-point (non-multiaccess) WAN interfaces
- You cannot reboot a IOS image over an ip unnumbered interface

Task: configure unnumbered interfaces on point-to-point WAN interfaces

```
(config-if)# ip unnumbered <interface-type interface-id>
```

31-bit prefix

- Conserve IP address space
- Since RFC 3021
- Only on point-to-point WAN interfaces

Task: Use a 31-bit prefix on point-to-point WAN interfaces

```
(config)# ip classless
(config-if)# ip address a.b.c.d 255.255.255.254
```

10.2. ARP

- **Configuration Guides | IP Addressing | ARP**
- RFC 826
- Finds MAC address of a host given IP address
- maintains ARP cache
- ARP Ethernet protocol: 0x0806

10.2.1. Protocol

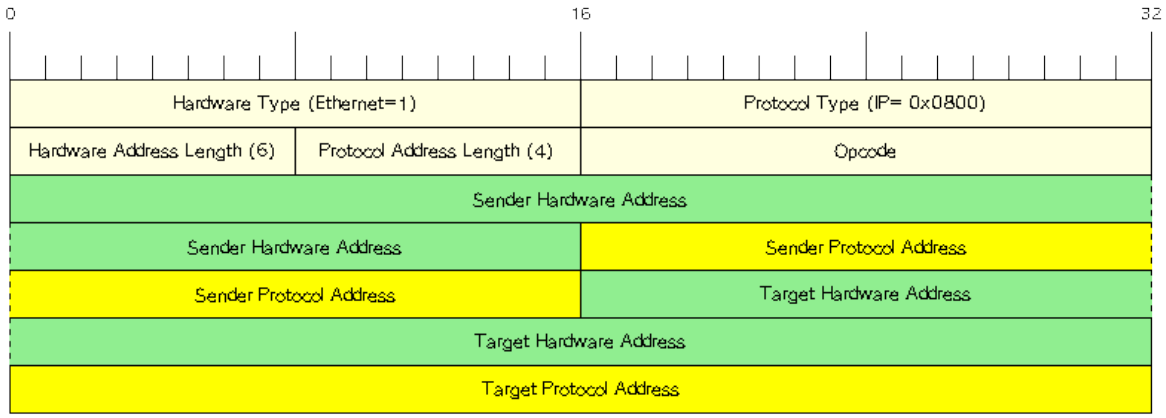


Figure 12. header format

OpCode

- 1 for request
- 2 for response

10.2.2. Static ARP entries

Task: Enable the interface encapsulation

```
(config-if)# arp {arpa | frame-relay | snap}
```

Task: Define static ARP entries

```
# arp <ip-address>  <hardware-address> <encapsulation-type> [<interface-type>]
```

Task: Define static ARP entries for a specific VRF

```
# arp vrf <name>  <hardware-address> <encapsulation-type> [<interface-type>]
```

10.2.3. Dynamic ARP entries

- Stored in ARP cache
- timeout by default in 4 hours

Task: clear the entire ARP cache on an interface

```
# clear arp interface <type-number>
```

Task: clear all dynamic entries from the ARP cache, the fast-switchig cache, and the IP route cache

```
# clear arp-cache
```

Task: set an expiration time for dynamic entries in the arp cache

```
(config-if)# arp timeout <seconds>
```

Task: Display the arp cache

```
# show ip arp
```

```
# show arp

Protocol    Address           Age (min)   Hardware Addr      Type    Interface
Internet    10.108.42.112     120         0000.a710.4baf     ARPA    Ethernet3
AppleTalk   4028.5            29          0000.0c01.0e56     SNAP    Ethernet2
Internet    110.108.42.114    105         0000.a710.859b     ARPA    Ethernet3
AppleTalk   4028.9            -           0000.0c02.a03c     SNAP    Ethernet2
Internet    10.108.42.121     42          0000.a710.68cd     ARPA    Ethernet3
Internet    10.108.33.9       -           0000.0c01.7bbd     SNAP    Fddi0
```

Task: Display ARP and RARP processes

```
# sh processes cpu | include (ARP|PID)
```

10.2.4. Proxy ARP

- Same message as ARP
- Allows response for IP address in remote subnet.
- RFC 1027

- Mostly replaced by DHCP nowadays

Task: disable Proxy ARP globally

```
# ip arp proxy disable
```

Task: Disable Proxy ARP on an interface

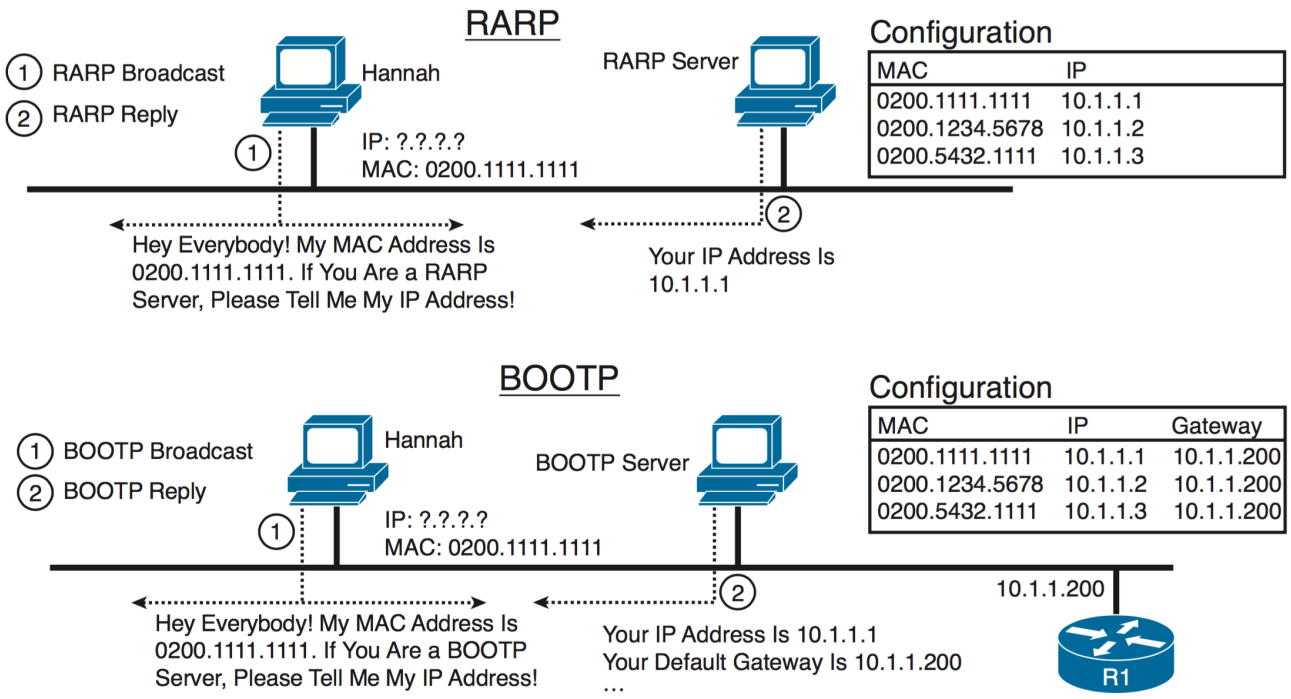
```
(config-if)# no ip proxy-arp
```

10.2.5. Gratuitous ARP

- broadcast ARP messages where the SPA=TPA and THA=FF:FF:FF:FF:FF:FF
- used to
  - to detect IP address conflit
  - to update other machine ARP table
  - to update mac table of the connected switche

If we see multiple gratitous ARPs from the same host frequently, it can be an indication of bad Ethernet NICs

10.2.6. RARP and BootP as DHCP precursors



- A RARP request is a host's attempt to find its own IP address. So RARP uses the same old ARP message, but the ARP request lists a MAC address target of its own MAC address and a target IP address of 0.0.0.0. A preconfigured RARP server, which must be on the same subnet as the

client, receives the request and performs a table lookup in its configuration. If that target MAC address listed in the ARP request is configured on the RARP server, the RARP server sends an ARP reply, after entering the configured IP address in the Source IP address field.

- BOOTP was defined in part to improve IP address assignment features of RARP. BOOTP uses a completely different set of messages, defined by RFC 951, with the commands encapsulated inside an IP and UDP header. With the correct router configuration, a router can forward the BOOTP packets to other subnets—allowing the deployment of a centrally located BOOTP server. Also, BOOTP supports the assignment of many other tidbits of information, including the subnet mask, default gateway, DNS addresses, and its namesake, the IP address of a boot (or image) server. However, BOOTP does not solve the configuration burden of RARP, still requiring that the server be preconfigured with the MAC addresses and IP addresses of each client.

10.3. DHCP

Configuration guides | IP Addressing | DHCP

- RFC 2131
  - Dynamic Host Configuration Protocol
  - Based on BOOTP (itself based on RARP)
  - Client/agent relay/server model
  - UDP port 67

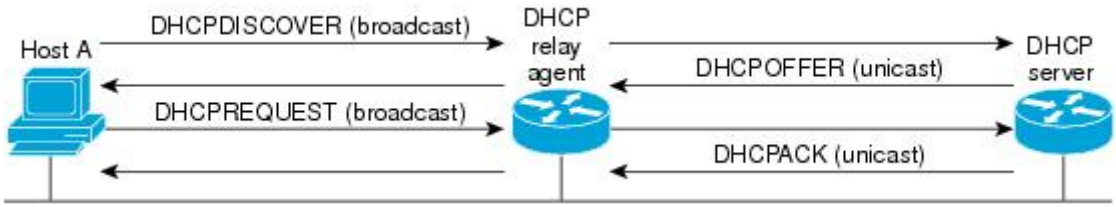


Figure 13. DHCP request for an IP address from a DHCP Server

10.3.1. Protocol operations

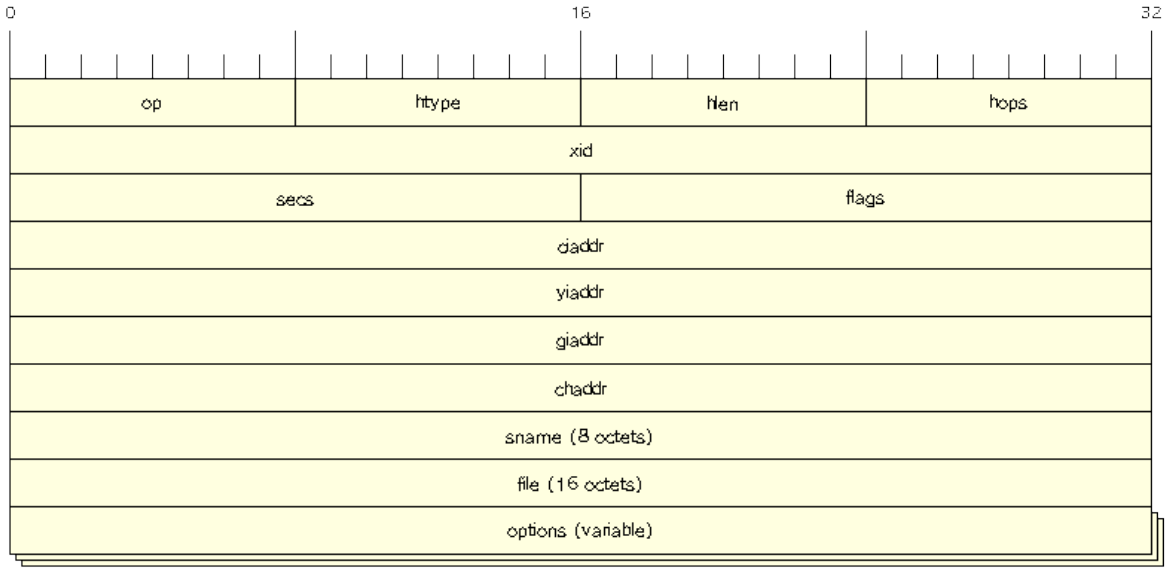


Figure 14. DHCP message

FIELD	OCTETS	DESCRIPTION
op	1	Message op code / message type. 1 = BOOTREQUEST, 2 = BOOTREPLY
htype	1	Hardware address type
hlen	1	Hardware address length
hops	1	Client sets to zero, optionally used by relay agents when booting via a relay agent.
xid	4	Transaction ID, a random number chosen by the client, used by the client and server to associate messages and responses between a client and a server.
secs	2	Filled in by client, seconds elapsed since client began address acquisition or renewal process.
flags	2	Flags
ciaddr	4	Client IP address; only filled in if client is in BOUND, RENEW or REBINDING state and can respond to ARP requests.
yiaddr	4	'your' (client) IP address.
siaddr	4	IP address of next server to use in bootstrap returned in DHCPOFFER, DHCPACK by server.
giaddr	4	Relay agent IP address, used in booting via a relay agent.
chaddr	16	Client hardware address.
sname	64	Optional server host name, null terminated string.
file	128	Boot file name, null terminated string; "generic" name or null in DHCPDISCOVER, fully qualified directory-path name in DHCPOFFER.
options	var	Optional parameters field.

10.3.2. DHCP Server

- Accepts address assignment requests and renewals from clients
- Assign address, name server, gateways, ...
- Accepts broadcasts from local clients or relay agents
- Database as a tree used for attribute inheritance
  - Root: address pool for natural networks
  - Branches: subnetwork address pools
  - Leaves: manual bindings

Task: Clear DHCP server variables

```
clear ip dhcp binding { <address> | * }
clear ip dhcp conflict { <address> | * }
clear ip dhcp server statistics
```

Task: Troubleshoot DHCP IP address assignments, lease expirations, and database chnages

```
# debug ip dhcp server events
```

Database agent

- Host (ftp, tftp, rcp) or storage that stores the DHCP bindings database.

Task: Save automatic bindings on a remote host

```
ip dhcp database <url> [timeout <seconds>] [ write-dely <seconds>]
```

- **url**: can be ftp,tftp, rcp, flash, disk
- **timeout**: how long the DHCP server wait before aborting database transfer. default: 5 minutes
- **write-delay**: how soon the DHCP server should send database updates. default: 5 minutes, minimum: 60 seconds

Task: Run DHCP server without database agent

```
(config)# no ip dhcp conflict logging
```

- Not recommended
- TODO: add the reason

Address Pool

- Specify which DHCP options to use for the client
  - If the client is not directly connected to the DHCP server (the giaddr field of the DHCPDISCOVER broadcast message is nonzero), the server matches the DHCPDISCOVER with the DHCP pool that has the subnet that contains the IP address in the giaddr field.
  - If the client is directly connected to the DHCP server (the giaddr field is zero), the DHCP server matches the DHCPDISCOVER with DHCP pools that contain the subnets configured on the receiving interface. If the interface has secondary IP addresses, subnets associated with the secondary IP addresses are examined for possible allocation only after the subnet associated with the primary IP address (on the interface) is exhausted.

Task: Create a pool

```
(config)# ip dhcp pool <name>
```

Task: Specify the subnet network number and mask of the address pool

```
(dhcp-config)# network <network-number> [mask | prefix-length]
```

Task: Specify the secondary subnets

```
(dhcp-config)# network <network-number> [mask | prefix-length] secondary
```

Task: Exclude IP address

```
(config)# ip dhcp excluded-address <low-address> [<high-address>]
```

Task: Specify the domain name

```
(dhcp-config)# domain-name <example.com>
```

Task: Specify the name server per order of preference

```
(dhcp-config)# dns-server <address> [<address2> ... <address8>]
```

Task: Specify the default boot image for a client

```
(dhcp-config)# bootfile <filename>
```

Task: Specify the netbios server

```
(dhcp-config)# netbios-name-server <address> [<address2> ... <address8>]
(dhcp-config)# netbios-node-type <type>
```

Task: Specify the gateway

```
(dhcp-config)# default-router  <address> [<address2> ... <address8>]
```

Task: Specify a custom DHCP code

```
(dhcp-config)# option <code> [instance <number>] {ascii <string> | hex <string> | <ip-
address>}
```

Task: Configure the duration of the lease

```
(dhcp-config)# lease <days> [<hours> [<minutes>] ]
```

Task: Specify the lease for ever

```
(dhcp-config)# lease infinite
```

Task: Configure the utilization mark of the current address pool size

```
(dhcp-config)# utilization mark high <percentage-number> [log]
(dhcp-config)# utilization mark low <percentage-number> [log]
```

Task: Configure a DHCP address pool with secondary subnets

```
(dhcp-config)# override default-router ??
(dhcp-config)# override utilization high <percentage>
(dhcp-config)# override utilization low <percentage>
```

TODO: add explanation

Task: Verify the DHCP address pool configuration

```
# show ip dhcp pool [name]
# show ip dhcp binding [address]
# show ip dhcp conflict [name]
# show ip dhcp database [url]
# show ip dhcp server statistics [type-number]
```

Address bindings

- Mapping between the IP address and MAC address of a client

Task: Display the current mapping

```
# show ip dhcp binding
```

automatic bindings

- Dynamically maps hardware address to an IP address from a pool.
- Stored in volatile RAM and periodically copied to database agent

manual binding

- MAC address of hosts are found in the DHCP database
- Stored in NVRAM
- Can be configured
  - Individually and stored in NVRAM
  - In batch from text files

Task: Specify the IP address and subnet mask of the client

```
(dhcp-config)# host <address> [<mask>| </prefix-length]
```

Task: Specify the unique identifier for a DHCP client

```
(dhcp-config)# client-identifier <unique-identifier>
```

- Send with DHCP option 61
- Unique identifier
  - 7-byte: 1byte for the media , 6 byte for the MAC address
  - 27-byte: vendor, MAC address, source interface of the client

Task: Determine the client identifier

```
# debug ip dhcp server packet

DHCPD:DHCPDISCOVER received from client 0b07.1134.a029 through relay 10.1.0.253.
DHCPD:assigned IP address 10.1.0.3 to client 0b07.1134.a029.
```

Task:

```
(dhcp-config)# hardware-address <hw-address> [<protocol-type> | <hw-number>]
```

- For client who can not send a client identifier in the packet

Task:

```
(dhcp-config)# client-name <name>
```

- Do not include the domain name

Static mapping

- From customer-created text file that DHCP server reads at boot
  - Short configuration: no need for several numerous host pools with manual bindings
  - Reduce space required in NVRAM to maintain address pools
- The file format has the following elements:
  - Database version number
  - End-of-file designator
  - Hardware type
  - Hardware address
  - IP address
  - Lease expiration
  - Time the file was created

Example

```
*time* Jan 21 2005 03:52 PM
*version* 2
!IP address    Type    Hardware address     Lease expiration
10.0.0.4 /24   1       0090.bff6.081e       Infinite
10.0.0.5 /28   id      00b7.0813.88f1.66    Infinite
10.0.0.2 /21   1       0090.bff6.081d       Infinite
*end*
```

Task: Configure the DHCP server to read a static mapping text file

```
(dhcp-config)# origin file <url>
```

Pings

- DHCP server pings an IP address twice before assigning it to a client.
- If the ping is unanswered after waiting for 2 seconds, the server assumes that the address is not in use.

Task: Specify the number of packets sent to a pool address before assigning it to a client

```
(config)# ip dhcp ping packets <number>
```

Task: Specify how long a DHCP server waits for a ping reply from an address pool

```
(config)# ip dhcp ping timeout <milliseconds>
```

BOOTP interoperability

Task: Configure the DHCP server to not reply to any BOOTP requests.

```
(config)# ip dhcp boot ignore
```

Task: Forward ignored BOOTP request packets to another DHCP server

```
(config)# ip helper-address <a.b.c.d>
```

Central DHCP server

- Updates specific DHCP options for remote DHCP server

Task: Import DHCP option parameters from central DHCP server

```
(dhcp-config)# import all
(config)# interface <type> <number>
(config-if)# ip address dhcp
```

Task: Display the options that are imported from the central DHCP server

```
# sh ip dhcp import
```

Option 82

- DHCP option contains information known by the relay agent
- For dynamic IP addresses allocation
- TOBECOMPLETED
- By default, OS DHCP server uses info provided by option 82

Task: Enable DHCP address allocation with option 82

```
(config)# ip dhcp use class
```

Task: Define a DHCP class and relay agent information patterns

```
(config)# ip dhcp class <name>
(dhcp-class)# relay agent information
(dhcp-class-info)# relay-information hex <pattern> [*] [bitmask <mask>]
```

Task: Display DHCP class matching results

```
# debug ip dhcp server class
```

Static route with the next-hop dynamically obtained through DHCP

TODO: explanation/context

Task: Assign a static route for the default next-hop device when the DHCP server is accessed for an IP address

```
# ip route <prefix> <mask> {<ip-address> | <interface-number> [<ip-number>]} dhcp
[<distance>]
```

- Ensure that the DHCP client and server are defined to supply a DHCP device option 3 of the DHCP packet.
- If the DHCP client is not able to obtain an IP address or the default device IP address, the static route is not installed in the routing table.
- If the lease has expired and the DHCP client cannot renew the address, the DHCP IP address assigned to the client is released and any associated static routes are removed from the routing table.

Statistics

Task: Display server statistics

```
# show ip dhcp server statistics
```

Task: Reset all DHCP server counters to 0

```
# clear ip dhcp server statistics
```

10.3.3. DHCP Relay Agent

- Forwards requests and replies between clients and servers not on the same physical subnet
- Sets the **giaddr** field and adds option 82
- DHCP server and relay agent are enabled by default

Task: Specify the packet forwarding address

```
(config-if)# ip helper-address <a.b.c.d>
```

Task: Reduce the frequency with which DHCP clients change their addresses and forwards client requests to the server that handle the previous request.

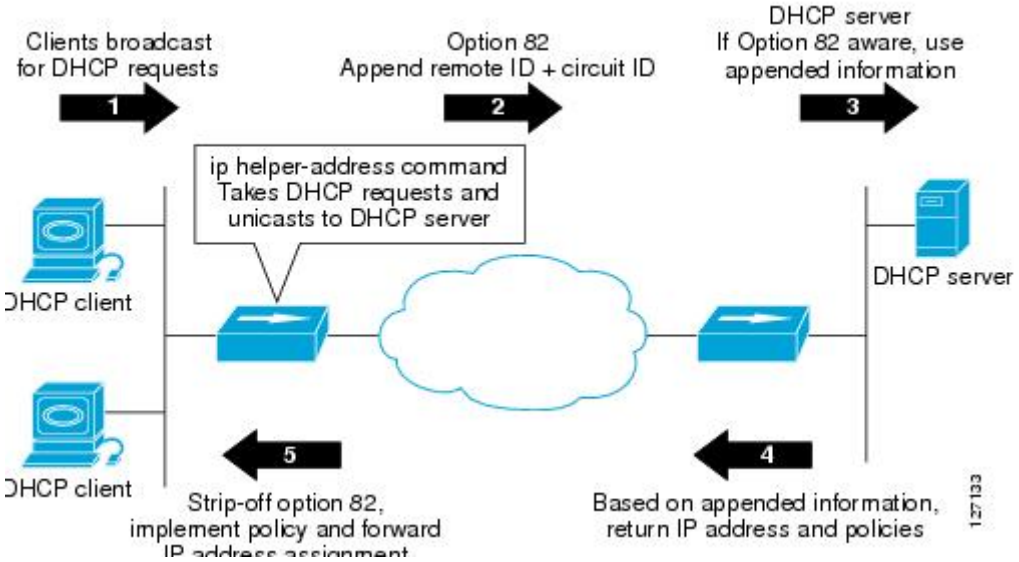
```
(config-if)# ip dhcp relay prefer known-good-server
```

- The relay agent deletes the ARP entries for addresses offered to the client on unnumbered interfaces.

Task: Disable the DHCP relay agent service

```
# no service dhcp
```

Option 82



Task: Insert the DHCP relay agent information option in BOOTREQUEST messages forwarded to a DHCP server

```
# ip dhcp relay information option
```

- This function is disabled by default

Task: Check whethers the relay agent information option forwarded BOOTREPLY message is valid

```
# ip dhcp relay information check
```

Task: Configure the reforwarding policy

```
# ip dhcp relay information policy {drop | keep | replace }
```

Task: Configure all interfaces as trusted sources of the DHCP relay information option.

```
# ip dhcp relay information trust-all
```

Task: Configure an interface as trusted sources of the DHCP relay information option.

```
(config-if)# ip dhcp relay information trusted
```

Task: Display all interfaces that are configure to be a trusted source for the DHCP relay information option.

```
# show ip dhcp relay information trusted-sources
```

Task: Configure per-interface support for the relay agent information option

```
(config-if)# ip dhcp relay information option-insert [none]
(config-if)# ip dhcp relay information check-reply [none]
(config-if)# ip dhcp relay information policy-action {drop | keep | replace}
```

See more optional tasks here

10.3.4. DHCP Client

Task: Acquire an IP address on an interface from DHCP

```
(config-if)# ip address dhcp
```

Task: Display the DHCP packets sent and received during troubleshooting on the client side

```
# debug dhcp detail
```

Task: Force a release of a DHCP lease

```
# release dhcp
```

The **release dhcp** command

- Starts the process to immediately release a DHCP lease for the specified interface.
- Does not deconfigure the **ip address dhcp** command specified in the configuration file for the interface.

Task: Force a renewal of a DHCP lease

```
# renew dhcp
```

- The **renew dhcp** command advances the DHCP lease timer to the next stage, at which point one of the following occurs:
  - If the lease is currently in a BOUND state, the lease is advanced to the RENEW state and a DHCP RENEW request is sent.
  - If the lease is currently in a RENEW state, the timer is advanced to the REBIND state and a DHCP REBIND request is sent.
- If there is no response to the RENEW request, the interface remains in the RENEW state. In this case, the lease timer will advance to the REBIND state and subsequently send a REBIND request.
- If a NAK response is sent in response to the RENEW request, the interface is deconfigured.

Configurable DHCP client feature

- Allows a client to use a user-specified client identifier, class identifier or suggested lease time when requesting an address from a DHCP server.
- Options available:
  - Option 33: configure a list of static routes in the client.
  - Option 51: request a lease time for the IP address.
  - Option 55: request certain options from the DHCP server
  - Option 60: configure the vendor class identifier string to use in the DHCP interaction.
  - Option 61: specify their unique identifier

FORCERENEW Message Handling

TODO: Explain the feature

Task: Configure FORCERENEW message handling

```
! Specify the key chain to be used in authenticating a request
(config)# key chain <name>
(config-keychain)# key <id>
(config-keychain-key)# key-string <text>
!
! Specify the type of authentication
(config)# interface <type number>
(config-if)# ip dhcp client authentication key-chain <name>
(config-if)# ip dhcp client authentication mode <type>
!
# ip dhcp-client forcerenew
```

10.3.5. Accounting and Security

- Address vulnerability in PWLAN

DHCP Accounting

- add AAA and RADIUS support to DHCP configuration
- sends secure START/STOP accounting messages upon lease assignment/termination
- Restrictions:
  - AAA and RADIUS must be enabled
  - only for network pools with automatic bindings
  - **clear ip dhcp binding** or **no service dhcp** triggers accounting STOP messages

Task: Enable DHCP accounting if a specifier server group is configured to run RADIUS accounting

```
(dhcp-config)# accounting <method-list-name>
```

Task: Troubleshoot DHCP accounting

```
debug radius accounting
debug ip dhcp server events
debug aaa accounting
debug aaa id
```

DHC secured IP address assignment

- Secures and synchronizes the MAC address of the client to the DHCP binding, preventing hackers form spoofing the DHCP server and taking over a DHCP lease of an authorized client

Task: Secure ARP table entries to DHCP leases in the DHCP database

```
(dhcp-config)# update arp
```

- Existing active DHCP leases will not be secured until they are renewed.

Task: Configure the renewal policy for unknown clients

```
(dhcp-config)# renew deny unknown
```

- In some usage scenarios, such as a wireless hotspot, where both DHCP and secure ARP are configured, a connected client device might go to sleep or suspend for a period of time. If the suspended time period is greater than the secure ARP timeout (default of 91 seconds), but less than the DHCP lease time, the client can awake with a valid lease, but the secure ARP timeout has caused the lease binding to be removed because the client has been inactive. When the client awakes, the client still has a lease on the client side but is blocked from sending traffic. The client will try to renew its IP address but the DHCP server will ignore the request because the DHCP server has no lease for the client. The client must wait for the lease to expire before being able to recover and send traffic again.
- To remedy this situation, use the **renew deny unknown** command in DHCP pool configuration mode. This command forces the DHCP server to reject renewal requests from clients if the requested address is present at the server but is not leased. The DHCP server sends a DHCPNAK denial message to the client, which forces the client back to its initial state. The client can then negotiate for a new lease immediately, instead of waiting for its old lease to expire.

DHCP per interface lease limit and statistics

- Allows an ISP to limit the number of DHCP leases allowed on an interface.

Task: Configure a DHCP lease limit to control the number of subscribers on an interface

```
(config)#  ip dhcp limit lease log
(config-if)# ip dhcp limit lease <max-users>
```

Task: Verify the DHCP lease limit configuration

```
# show ip dhcp limit lease
```

Task: Clear the stored lease violation entries

```
# clear ip dhcp limit lease
```

DHCP authorized ARP

Task: Disable dynamic ARP learning on an interface

```
(config-if)# arp authorized
```

Task: Configure how long an entry remains in the ARP cache

```
(config-if)# arp timeoute <seconds>
```

Task:

```
# show arp
```

ARP auto-logoff

- enhances DHCP authorized ARP by providing finer control and probing authorized clients to detect a logoff.

Task: Configure an interval and number of probe retries for ARP

```
(config-if)# arp probe interval <seconds> count <number>
```

DHCP snooping

TODO

add information about option 82

10.4. NAT

- **Configuration Guides | IP Addressing | NAT**
- RFC 1631

10.4.1. Purpose

- IPv4 address conservation
- can be static, dynamic or pat

10.4.2. Inside and outside address

Inside local address

The (private) IP address that is assigned to a host on the inside network.

Inside global address

A (public) IP address that represents one or more inside local IP addresses to the outside world.

Outside local address

The (private) IP address of an outside host as it appears to the inside network.

Outside global address

The (public) IP address assigned to a host on the outside network by the owner of the host.

Task: Display NAT translation information

```
show ip nat translations [verbose]
show ip nat statistics
```

10.4.3. Static NAT

- Statically correlates the same local host to the same public IP address.
- Does not conserve IP addresses.

Task: Configure static translation of inside source address

```
conf t
ip nat inside source static <local-ip> <global-ip>

interface <type number>
  ip address <ip-address> <mask> [secondary]
  ip nat inside

interface type number
  ip address <ip-address> <mask>
  ip nat outside
```

10.4.4. Dynamic NAT without PAT

- One local host uses an available public IP address in a pool.
- Does not conserve IP addresses.
- Timeout after period of nonuse

Task: Configure dynamic translation of inside source address

```
ip nat pool <name> <start-ip> <end-ip> {netmask <mask> | prefix-length <length>}
access-list <acl> permit source [<w.i.l.d>]
ip nat inside source list <acl> pool <name>

interface <type number>
  ip address <ip-address> <mask>
  ip nat inside

interface <type number>
  ip address <ip-address> <mask>
  ip nat outside
```

Task: Change timeouts value

```
ip nat translation <seconds>
ip nat translation udp-timeout <seconds>
ip nat translation dns-timeout <seconds>
ip nat translation tcp-timeout <seconds>
ip nat translation finrst-timeout <seconds>
ip nat translation icmp-timeout <seconds>
ip nat translation syn-timeout <seconds>
```

10.4.5. PAT

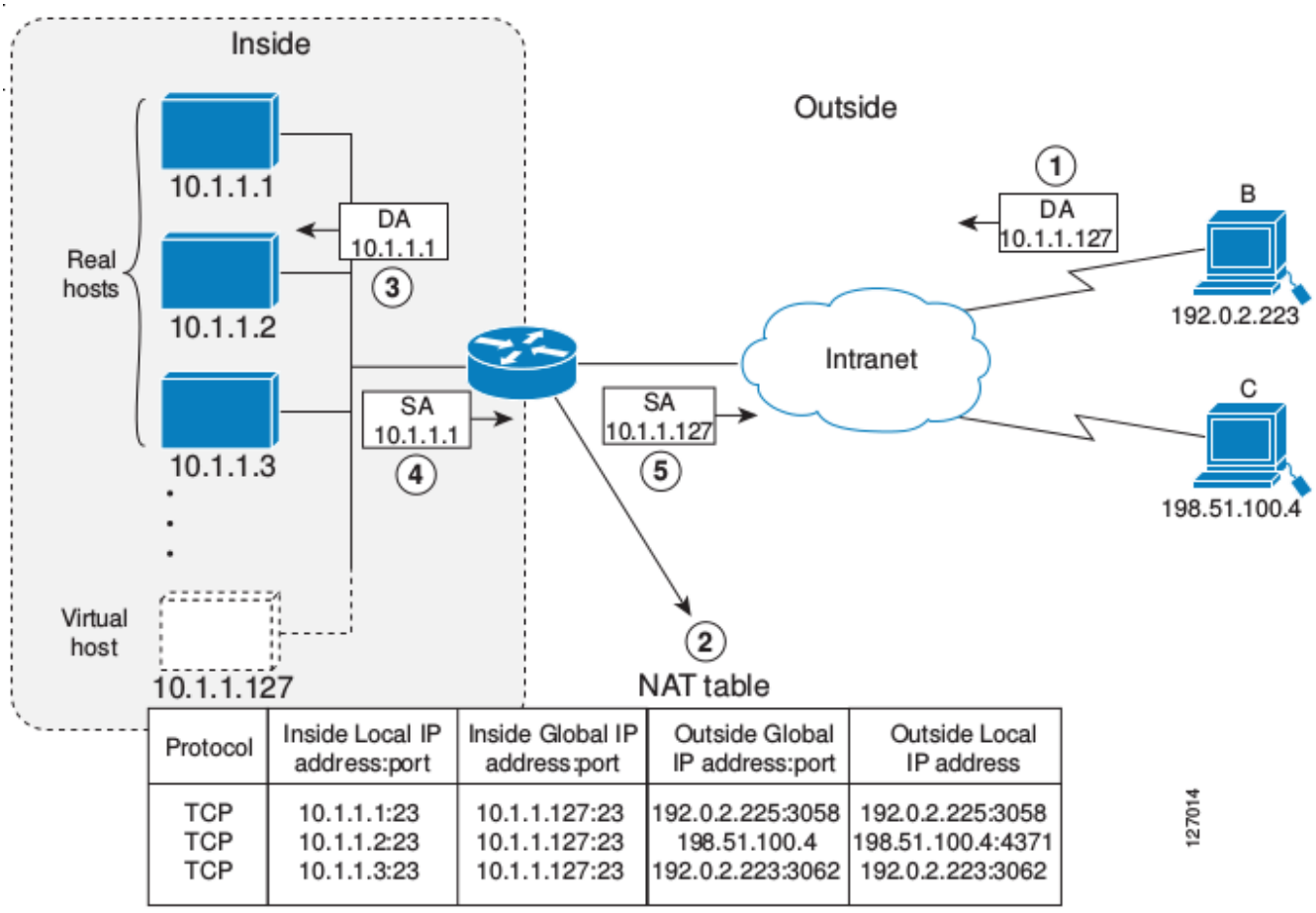
- Like dynamic NAT but multiple local hosts share a single public address by multiplexing TCP/UDP ports.
- Conserves IP addresses.

10.4.6. NAT for overlapping address

- Can be done with any of the first three types.
- Translates both source and destination addresses, instead of just the source (for packets going from enterprise to the Internet).

10.4.7. TCP load distribution for NAT

- Round-robin allocation of a virtual host that coordinates load sharing among real hosts.



Protocol	Inside Local IP address:port	Inside Global IP address:port	Outside Global IP address:port	Outside Local IP address
TCP	10.1.1.1:23	10.1.1.127:23	192.0.2.225:3058	192.0.2.225:3058
TCP	10.1.1.2:23	10.1.1.127:23	198.51.100.4	198.51.100.4:4371
TCP	10.1.1.3:23	10.1.1.127:23	192.0.2.223:3062	192.0.2.223:3062

Figure 15. NAT TCP load distribution

Task: Allow internal users access to the internet

```
ip nat pool <name> <start-ip> <end-ip> {netmask netmask | prefix-length prefix-length}
access-list number permit a.b.c.d [e.f.g.h]
ip nat inside source list number pool name overload

interface type number
  ip address ip-address mask
  ip nat inside

interface type number
  ip address ip-address mask
  ip nat outside
end
```

10.4.8. Overlapping networks

Configure dynamic translation of overlapping networks if your IP addresses in the stub network are legitimate IP addresses belonging to another network and you want to communicate with those hosts or routers using dynamic translation.

Task: Configure overlapping network

```
ip nat pool name start-ip end-ip {netmask netmask | prefix-length prefix-length}
access-list access-list-number permit source [source-wildcard]
ip nat outside source list access-list-number pool name

interface type number
  ip address ip-address mask
  ip nat inside

interface type number
  ip address ip-address mask
  ip nat outside
```

10.4.9. Server TCP load balancing

```
ip nat pool name start-ip end-ip {netmask netmask | prefix-length prefix-length} type
rotary
access-list access-list-number permit source [source-wildcard]
ip nat inside destination-list access-list-number pool name

interface type number
  ip address ip-address mask
  ip nat inside

interface type number
  ip address ip-address mask
  ip nat outside
```

Task: Clear NAT entries before the timeout

```
clear ip nat translation inside global-ip local-ip outside local-ip global-ip
clear ip nat translation outside global-ip local-i p
clear ip nat translation protocol inside global-ip global-port local-ip local-port
outside local-ip local-port-global-ip global-port
clear ip nat translation {* | [forced] | [inside global-ip local-ip] [outside local-ip
global-ip]}
```

Task: Enable Syslog for logging NAT translations

```
ip nat log translations syslog
no logging console
```

10.4.10. NAT order of operations

Inside-to-Outside

1. If IPSec then check input access list
2. decryption - for CET (Cisco Encryption Technology) or IPSec
3. check input access list
4. check input rate limits
5. input accounting
6. redirect to web cache
7. policy routing
8. routing
9. NAT inside to outside (local to global translation)
10. crypto (check map and mark for encryption)
11. check output access list
12. inspect (Context-based Access Control (CBAC))

13. TCP intercept
14. encryption
15. Queueing

Outside-to-Inside

1. If IPSec then check input access list
2. decryption - for CET or IPSec
3. check input access list
4. check input rate limits
5. input accounting
6. redirect to web cache
7. NAT outside to inside (global to local translation)
8. policy routing
9. routing
10. crypto (check map and mark for encryption)
11. check output access list
12. inspect CBAC
13. TCP intercept
14. encryption
15. Queueing

Read more: Order of operations

10.5. NHRP

TIP: if dmvpn phase 3, the tunnel key must be the same as the tunnel key

!! DMVPN HUB int f0/0.123 enc dot1q 123 ip address 10.0.0.1 255.255.255.0 no shut int t123 ip add 129.99.123.1 255.255.255.0 tunnel source f0/0.123 tunnel mode gre multipoint tunnel key 123 ip nhrp network-id 123 ip nhrp map multicast dynamic ip nhrp network-id 321

!! DMVPN SPOKE int f0/0.123 desc ospf enc dot1q 123 ip address 10.0.0.2 255.255.255.0 int t123 ip add 129.99.123.2 255.255.255.0 tunnel source f0/0.123 tunnel destination 10.0.0.1 tunnel key 123 ip nhrp network-id 123 ip nhrp nhs 129.99.123.1 ip nhrp map multicast 10.0.0.1 ip nhrp map 129.99.123.1 10.0.0.1

Task: Verify that NHRP registration has been sent from spokes to the hub

```
R1#sh ip nhrp

129.99.123.2/32 via 129.99.123.2
   Tunnel123 created 00:08:18, expire 01:54:55
   Type: dynamic, Flags: unique registered
   NBMA address: 10.0.0.2
129.99.123.3/32 via 129.99.123.3
   Tunnel123 created 00:09:22, expire 01:54:57
   Type: dynamic, Flags: unique registered
   NBMA address: 10.0.0.3
```

10.6. IPv6

10.6.1. IPv6 Header

Table 8. IPv6 Base Header Format

Comparison with IPv4	Packet
Streamlined:: - Fragmentation fields moved out of base header - IP options moved out of base header - Header checksum eliminated - Header length field eliminated - Length field excludes IPv6 header - Alignment changed from 32 to 64 bits Revised:: - Time to live → hop limit - Protocol → next header - Precedence and TOS → traffic class - Addresses increased 32 bits → 128 bits Extended:: - Flow label field added	["packetdiag", target="ipv6-header-format",size=200] ---- diagram { colwidth = 32 node_height = 32 default_node_color = lightyellow default_fontsize = 12 * Version [len=4] * Traffic Class [len=8] * Flow Label [len=20] * Payload Length [len=16] * Next Header [len=8] * Hop Limit [len=8] * Source Address (128 bits) [len=32] * Destination Address (128 bits) [len=32] } ----

Traffic Class

- 6 bits for ToS
- 2 bits for ECN

Flow Label

- Sequence in a particular flow

Payload Length

-

Next header

When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order:

- IPv6 header
- Hop-by-hop options header
- Destination options header (routing header associations)
- Routing header
- Fragment header
- Authentication header
- Encapsulating security payload header
- Destination options header (options processed by final estination)
- Upper-layer header

Header Type	Next Header Value	Description
Hop-by-Hop Options Header	0	Read by all devices in transit network
Destination Option Header	60	Read by the final destination device
Routing Header	43	Support routing decision making
Fragment Header	44	Contains parameters of datagram fragmentation
Authentication Header	51	
Encapsulating Security Payload	50	Carries encrypted data for secure communication.
Upper-Layer Header	6	(TCP)
	Upper-Layer Header	17
(UDP)		Mobility Header (currently without upper layer)

- Value 59 means **No Next Header**

10.6.2. Addressing

- 128 bits
- Represented in hexadecimal and uses 8 colon-separated fields of 16 bits.

IPv4 vs IPv6

- Multiple ipv6 addresses on a logical or physical interface with equal precedence on IOS (only one primary ipv4 with optional secondary address)
- Automatic configuration of globally unique address (without the need of DHCP)
- Built-in neighbor discovery of neighbors, routers and gateways

Address abbreviation rules

- Whenever one or more successive 16-bit groups in an IPv6 address consist of all 0s, that portion of the address can be omitted and represented by two colons (::). The two-colon abbreviation can be used only once in an address, to eliminate ambiguity.
- When a 16-bit group in an IPv6 address begins with one or more 0s, the leading 0s can be omitted. This option applies regardless of whether the double-colon abbreviation method is used anywhere in the address.

```
2001:0001:0000:0000:00A1:0CC0:01AB:397A

2001:1:0:0:A1:CC0:1AB:397A
2001:0001::00A1:0CC0:01AB:397A
2001:1::A1:CC0:1AB:397A
```

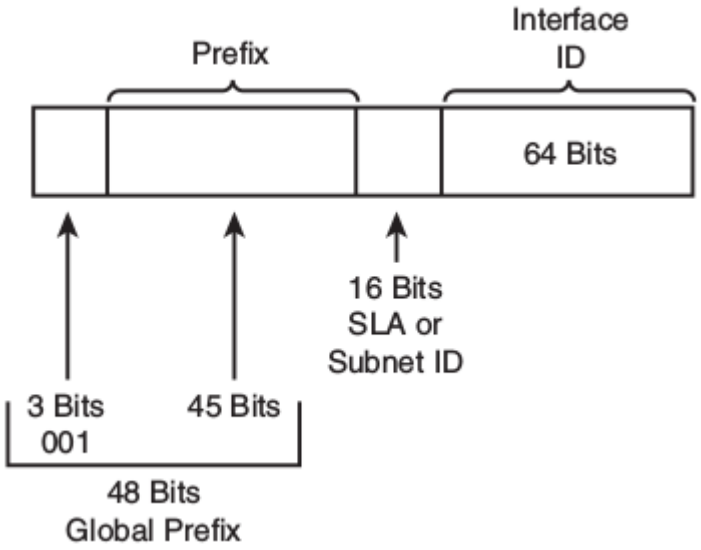
Address types

Address Type	Range	Application
Aggregatable global unicast	2000::/3	Host-to-host communication; same as IPv4 unicast.
Multicast	FF00::/8	One-to-many and many-to-many communication; same as IPv4 multicast.
Anycast	Same as Unicast	Application-based, including load balancing, optimizing traffic for a particular service, and redundancy. Relies on routing metrics to determine the best destination for a particular host.
Link-local unicast	FE80::/10	Connected-link communications.
Solicited-node multicast	FF02::1:FF00:0/104	Neighbor solicitation.

Unicast

Aggregatable global addresses

- Begin with binary 001 (hexadeximal= 2000::/3)



Link-local addresses

- Starts with FE80::/10
- Follows by 54 bits set to 0
- Interface ID
- Routers do not forward link-local traffic to other segments.

IPv4-compatible addresses

- One option is to have first 96 bits set to 0

```
0:0:0:0:0:10:10:100:16
::10:10:100:16
::A:A:64:10
```

Assign an IPv6 unicast address to a router interface

Task: Enable ipv6 on the router

```
(config)# ipv6 unicast-routing
```

Task: Configure a global unicast address

```
(config-if)# ipv6 address 2014:10:12::19:66/64
```

Router automatically configure a link local address on all IPv6 enabled interfaces. However, you can explicitly configure one

```
(config-if)# ipv6 address fe80::1 link-local
```

Multicast

IPv6 multicast address format

- Begin with FF as the first octect, or FF00::/8
- The second octet specifies lifetime (permanent or temporary) and the scope (node, link, site, organization, global)

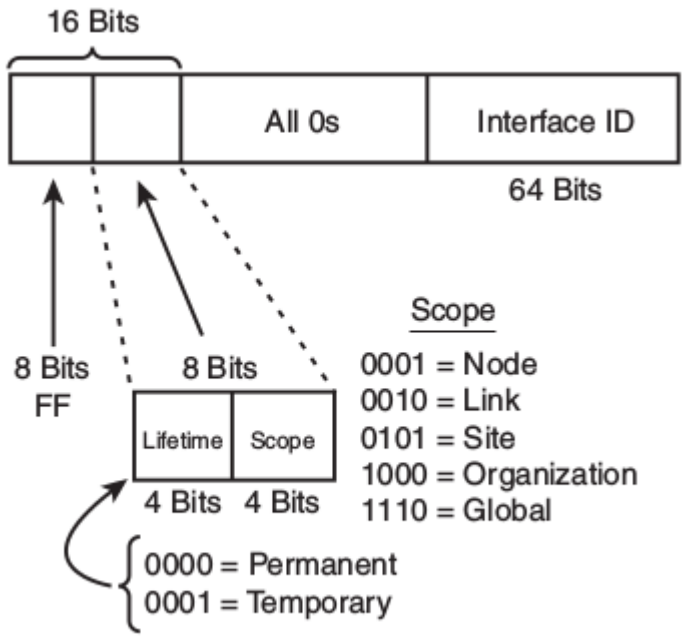


Table 9. IPv6 Multicast Well-Known Addresses

Function	Multicast Group	IPv4 Equivalent
All hosts	FF02::1	Subnet broadcast address
All Routers	FF02::2	224.0.0.2
OSPFv3 routers	FF02::5	224.0.0.5
OSPFv3 designated routers	FF02::6	224.0.0.6
EIGRP routers	FF02::A	224.0.0.10
PIM routers	FF02::D	224.0.0.13

Each router must join the **solicited-node group** (FF02::1:FF00:0000/104) for all unicast and anycast traffic. The last 24 bits come from the corresponding last 24 bits of the unicast or anycast address. The **neighbor discovery** process uses solicited-node addresses.

Anycast

Anycast addresses can be assigned to any number of hosts that provide the same service; when other hosts access this service, the specific server they hit is determined by the unicast routing metrics on the path to that particular group of servers. This provides geographic differentiation, enhanced availability, and load balancing for the service.

```
(config-if)# ipv6 address 3001:fffe::104/64 anycast
```

All IPv6 routers additionally must support the subnet router anycast address. This anycast address is a prefix followed by all 0s in the interface ID portion of the address. Hosts can use a subnet router anycast address to reach a particular router on the link identified by the prefix given in the subnet router anycast address.

The unspecified address

- Represented by **::**
- Used as source address by an interface that has not yet learned its unicast addresses.
- Cannot be assigned to an interface
- Cannot be used as a destination address

how to embed an RP address within a multicast group address

RFC 2373

Given address 2001:DB*:0717::A, Follow the structure FF7**X**:0**Y**30:2001:DB8:0717::**group**

- FF for a multicast address
- 7 indicates that the RP address is embedded in the multicast address
- X for the multicast scope
  - 1 node-local
  - 2 link-local
  - 5 site-local
  - 8 organization-local
  - E global
  - F reserved
- 0 in the first character of the second hextet
- Y for the RP interface ID from 1 to F
- 30 for the mask for the network (0x30 = decimal 48)
- Remaining hextets for the network prefix

IPv6 address autoconfiguration

Stateful autoconfiguration

- Assigns a host its entire 128-bit address using DHCP

Stateless autoconfiguration

- Assigns a host a 64-bit prefix, and the host derives the last bit using EUI-64 process.

EUI-64 address

- Split 48-bit MAC address in two parts
- Place FFFE in the middle
- Set to 1 the universal/local bit (7th bit in the interface id)

Given the IPv6 prefix 2001:128:1f:633 and MAC address 00:07:85:80:71:B8, the resulting EUI-address is 2001:128:1f:633:**2**07:85**FF:FE**80:71B8/64

```
(config-if)# ipv6 address 2001:128:1f:633::/64 eui-64
```

10.6.3. Basic IPv6 functionality protocols

Neighbor discovery

- RFC 2461
- Discover and track other IPv6 hosts on connected interfaces
- Uses ICMPv6 messages and Solicited-node multicast addresses
- Major roles
  - Stateless address autoconfiguration (detailed in RFC 2462)
  - Duplicate address detection (DAD)
  - Router discovery
  - Prefix discovery
  - Parameter discovery (link MTU, hop limits)
  - Neighbor discovery
  - Neighbor address resolution (replaces ARP, both dynamic and static)
  - Neighbor and router reachability verification

Neighbor advertisements

ICMPv6 messages used by ND

- Host advertises their pr
- Source addresses
- Destination addresses
- Icmp type, code: 134,0

Neighbor solicitation

- NS messages to find the link-layer of a specific neighbor
- Source address: manual assigned or ::
- Destination address: target address or solicited-node multicast address

- ICMP type, code: 135,0
- Uses in 3 operations: duplicate address detection, neighbor reachability verification, layer 3 to layer 2 address resolution.

IPv6 does not include ARP as a protocol but rather integrates the same functionality into ICMP as part of neighbor discovery. The response to an NS message is an NA message .

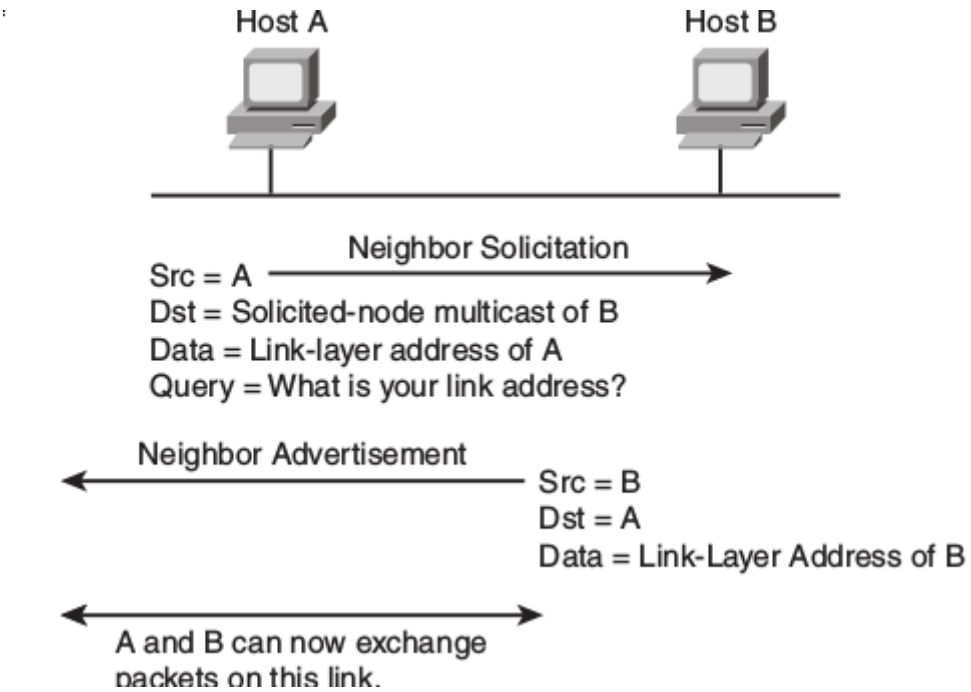


Figure 16. Neighbor discovery between two hosts

Router advertisement

- Routers advertise their presence and link prefixes, MTU, hop limits
- Source address: router's link-local address
- Destination address: all-nodes FF02::1 for periodic broadcasts, querying host address for response
- Icmp type, code: 134,0

A Cisco IPv6 router begins sending RA messages for each of its configured interface prefixes when the **ipv6 unicast-routing** command is configured. You can change the default RA interval (200 seconds) using the command **ipv6 nd ra-interval**. Router advertisements on a given interface include all of the 64-bit IPv6 prefixes configured on that interface. This allows for stateless address autoconfiguration using EUI-64 to work properly. RAs also include the link MTU, hop limits, and whether a router is a candidate default router.

IPv6 routers send periodic RA messages to inform hosts about the IPv6 prefixes used on the link and to inform hosts that the router is available to be used as a default gateway. By default, a Cisco router running IPv6 on an interface advertises itself as a candidate default router.

Task: Prevent router to advertise itself as a default candidate but do not hide its presence

```
ipv6 nd ra-lifetime 0
```

Task: Hide presence of a router running IPv6

```
ipv6 nd suppress-ra
```

Router solicitation

- Host query for the presence of routers on the link
- Source address: querying host interface, or :: if not assigned
- Destination address: FF02::2
- Icmp type, code : 133,0

At startup, IPv6 hosts can send RS messages to the all-routers multicast address. Hosts do this to learn the addresses of routers on a given link, as well as their various parameters, without waiting for a periodic RA message. If a host has no configured IPv6 address, it sends an RS using the unspecified address as the source. If it has a configured address, it sources the RS from the configured address.

Duplicate address Detection

To verify that autoconfigured or statically address is unique, the host sends an NS message to its own autoconfigured address's corresponding solicited-node multicast address. This message is sourced from the unspecified address ::. In the target address field in the NS is the address the host seeks to verify. If an NA from another host results, the sending host knows that the address is not unique

Neighbor unreachability detection

2 options:

- a host sends a probe to the desired host's solicited-node multicast address and receives an RA or an NA in response.
- a host, in communication with the desired host, receives a clue from higher-layer protocol (e.g. TCP ACK)

ICMPv6

- RFC 2463
- Two groups of messages: error reporting messages and informational messages
- IOS implements ICMP rate limiting by setting the minimum interval between error messages and build a token bucket

Limit ICMPv6 error messages with default interval 100 ms , and default token-bucket size 10.

```
(config)# ipv6 icmp error-interval seconds ???
```

Unicast reverse path forwarding

- Protects router from DoS attacks from spoofed IPv6 host address.
- Performs a recursive lookup in the ipv6 routing table to verify that the packet came in on the correct interface.

```
(config-if)# ipv6 verify unicast reverse-path
```

DNS

- Provides resolution of domain names
- DNS records: AAAA (RFC 1886), A6 (RFC 2874)

CDP

- Cisco Discovery Protocol
- Provides extensive information about the configuration and functionality of Cisco devices.

Task: Display IPv6 information transmitted in CDP

```
# show cdp neighbors detail
```

DHCP

- RFC 3315

Two conditions can cause a host to use DHCPv6:

- The host is explicitly configured to use DHCPv6 based on an implementation-specific setting.
- An IPv6 router advertises in its RA messages that it wants hosts to use DHCPv6 for addressing. Routers do this by setting the M flag (Managed Address Configuration) in RAs.

To use stateful autoconfiguration, a host sends a DHCP request to one of two well-known IPv6 multicast addresses on UDP port 547:

- FF02::1:2, all DHCP relay agents and servers
- FF05::1:3, all DHCP servers

The DHCP server then provides the necessary configuration information in reply to the host on UDP port 546. This information can include the same types of information used in an IPv4 network, but additionally it can provide information for multiple subnets, depending on how the DHCP server is configured.

To configure a Cisco router as a DHCPv6 server, you first configure a DHCP pool, just as in IPv4 then

enable the DHCPv6 service using the **ipv6 dhcp server pool-name**

Access lists

Similar with IPv4 access lists except that:

- Because Neighbor Discovery is a key protocol in IPv6 networks, access lists implicitly permit ND traffic. This is necessary to avoid breaking ND's ARP-like functionality. You can override this implicit-permit behavior using deny statements in IPv6 access lists.

Task: Configure an interface to filter traffic using an access list

```
ipv6 traffic-filter access-list-name {in | out}
```

- IPv6 access lists are always named; they cannot be numbered (unless you use a number as a name).
- IPv6 access lists are configured in named access-list configuration mode, which is like IPv4 named access-list configuration mode. However, you can also enter IPv4-like commands that specify an entire access-list entry on one line. The router will convert it to the correct configuration commands for named access-list configuration mode.

10.6.4. IPv6 routing

Static routes

Similar to IPv4 static routes except that:

- An IPv6 static route to an interface has an administrative distance of 1, not 0 as in IPv4.
- An IPv6 static route to a next-hop IP address also has an administrative distance of 1, like IPv4.
- Floating static routes work the same way in IPv4 and IPv6.
- An IPv6 static route to a broadcast interface type, such as Ethernet, must also specify a next-hop IPv6 address because
  - IPv6 does not use ARP
  - There is no concept of proxy ARP

```
(config)# ipv6 route 2001:128::/64 2001::207:85FF:FE80:7208
```

```
show ipv6 route
```

OSPFv3

implementing OSPF for IPv6

10.6.5. ospfv3

- Router id is highest ipv4 loopback, highest ipv4, or **router-id** id command

10.6.6. readings

Implement tunnels

Chapter 11. IP Switching

11.1. CEF

- enabled by default
- can be central or distributed accross multiple line cards
- Uses the FIB and the adjacency table
- Improves over process switching and fast switching methods

Switching Path	Forwarding Information stores	Load-Balancing Method
Process switching	Routing table	Per packet
Fast switching	Fast-switching cache (per flow route cache)	Per destination IP address
CEF	FIB tree and adjacency table	Per a hash of the packet source and destination

Task: disable CEF

```
(config)# no ip cef [distributed]
```

Task: disable CEF on an interface

```
(config-if)# no ip route-cache cef [distributed]
```

Task: Verify that CEF is enabled

```
# sh cef interface <type number> [detail]
# sh ip interface <type number>
```

11.1.1. FIB

- contains the prefixes and next-hop address from each entry in the IP routing table structured in a way that is optimized for forwarding.
- no need for route cache maintenance because there is a one-to-one correlation between FIB entries and routing table entries

Task: Display the FIB contents

```
# sh ip cef

Prefix              Next Hop            Interface
[...]
10.2.61.8/24        192.168.100.1       FastEthernet1/0/0
                    192.168.101.1       FastEthernet6/1
[...]
```

11.1.2. Adjacency Table

- stores outbound interface and MAC header rewrite for adjacent nodes

Task: Display the contents of the adjacency table

```
# show adjacency [detail]

Protocol       Interface            Address

IPV6           Serial0/0/0          point2point(12)
IP             Serial0/0/0          point2point(13)
IP             Serial0/0/1          point2point(15)
IPV6           Serial0/0/1          point2point(10)
IPV6           FastEthernet0/0.2    FE80:24::4(12)
...
```

Adjacency discovery

- adjacent nodes are discovered automatically (ARP) or added manually

Table 10. Adjacency Types that required special handling

Adjacency Type	Actions
Null adjacency	Packets destined for a Null0 interface are dropped. Null adjacency can be used as an effective form of access filtering.
Glean adjacency	When a device is connected to a multiaccess medium , the FIB table on the device maintains a prefix for the subnet rather than for the individual host prefixes. The subnet prefix points to a glean adjacency. A glean adjacency entry indicates that a particular next hop should be directly connected , but there is no MAC header rewrite information available. When the device needs to forward packets to a specific host on a subnet , Cisco Express Forwarding requests an ARP entry for the specific prefix , ARP sends the MAC address , and the adjacency entry for the host is built.
Punt adjacency	The device forwards packets requiring special handling or packets sent by features not yet supported in CEF switching paths to the next higher switching level for handling.
Discard adjacency	The device discards the packets.

Adjacency Type	Actions
Drop adjacency	The device drops the packets.

Unresolved Adjacency

When a link-layer header is prepended to a packet, the FIB requires the prepended header to point to an adjacency corresponding to the next hop. If an adjacency was created by the FIB and not discovered through a mechanism such as ARP, the Layer 2 addressing information is not known and the adjacency is considered incomplete or unresolved. Once the Layer 2 information is known, the packet is forwarded to the RP, and the adjacency is determined through ARP. Thus, the adjacency is resolved.

11.1.3. CEF load balancing

- per-destination (default)
- per-packet: round-robin method over multiple links

Task: disable per-destination

```
(config-if)# no ip load-sharing per-destination
```

Task: enable per-packet load balancing

```
(config-if)# ip load-sharing per-packet
```

Original algorithm

- produces distortions in load sharing across multiple routers because the same algorithm is used on every router.

Universal algorithm

- allows each router on the network to make a different load sharing decision for each source-destination address pair,
- avoids original CEF polarization
- Use a Universal ID as seed for the hash function
- default

Tunnel algorithm

- when there are only a few source and destination pairs

Include-ports algorithm

- allows you to use the Layer 4 source and destination ports as part of the load-balancing decision.
- benefits traffic streams running over equal cost paths that are not load shared because the majority of the traffic is between peer addresses that use different port numbers, Real-Time Protocol (RTP) streams.

GTP-U TEID-Based ECMP Load-Balancing Algorithm

- for Cisco IOS XE Software
- for mobile devices

Task: Select CEF load balancing algorithm

```
(config)# ip cef load-sharing algorithm {original | universal | tunnel | include-ports
[source | destination | source destination] }
```

Chapter 12. IP routing

12.1. RIP

Configuration Guides | IP Routing | RIP

- Distance vector protocol
- Transport: UDP 520
- Update destination:
  - Broadcast 255.255.255.255 for RIPv1
  - Multicast 224.0.0.9 for RIPv2
- Full updates every 30 seconds
- Trigerred updates
- Multiple routes to the same subnet with equal metric:
  - Default = 4
  - Configured with **ip maximum-paths *n***
- Metric: hop count with
  - 1 signifying a directly connected network of the advertising router
  - 16 signifying an unreachable network.
- AD: 120
- Support CIDR, VLSM, authentication
- Periodic updates every 30 seconds to multicast address 224.0.0.9
- Split horizon (without poison reverse on Cisco)
- Subnet mask included in route entry
- Administrative distance: 120
- Route tags when routes are redistributed into RIP
- Can advertise a next-hop router that is different from itself
  - Not implemented in Cisco IOS
- Does not keep a separate topology table
- Does not form neighbor relationship

12.1.1. RIP Messages

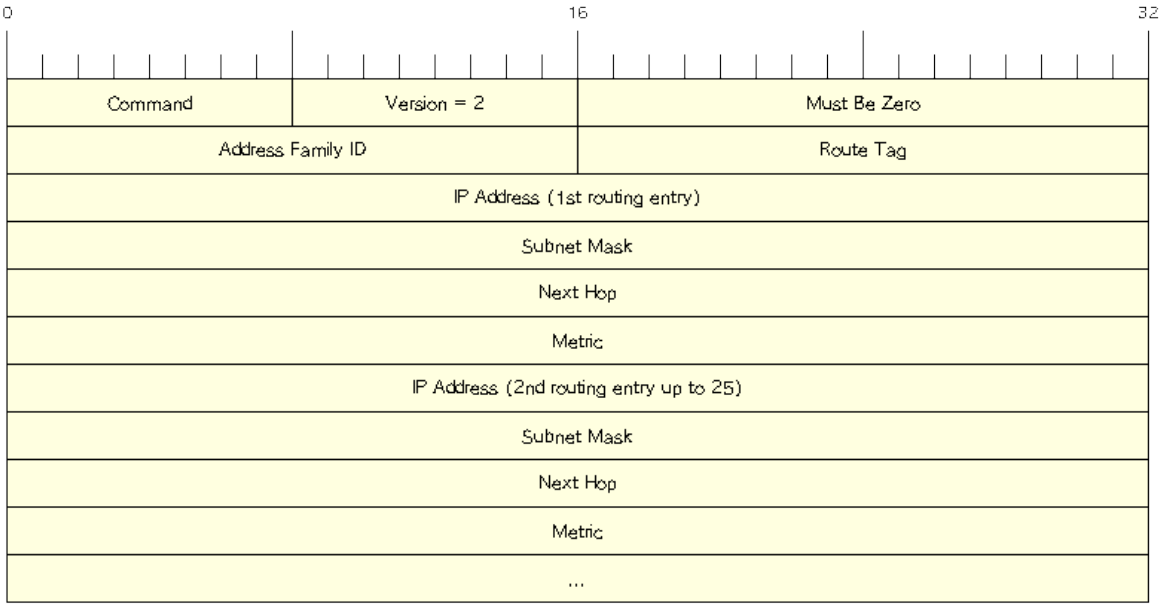


Figure 17. RIP header format

12.1.2. Request Message

- Command value = 1
- ask a neighbor to send a partial or a full RIP update immediately,
- Do not wait for the Update timer to expire
- Full RIP update
  - if one routing entry with AF = 0 and metric = 16
  - sent when RIP process start, RIP-enabled interfaces comes up, or **clear ip route**
- Partial update
  - if one or more route entry
  - Seldom used in Cisco IOS

12.1.3. Response Message

- Command value = 2

12.1.4. Default RIP configuration

- Version : 1
- Auto-summary : enable
- Authentication : disable
- Authentication mode: text
- Split-horizon : enable

- Interpacket delay : no

12.1.5. Basic configuration

```
(config)#router rip
(config-router)#version 2
(config-router)#network 10.0.0.0
(config-router)#no auto-summary
```

12.1.6. Version

Task: Specify the RIP version globally

```
(config-router)# version {1 | 2}
```

Task: Configure an interface to send only a RIPv2 packets

```
(config-if)ip rip send version [1] [2]
```

Task: Configure an interface to receive only a RIPv2 packets

```
(config-if)ip rip receive version [1] [2]
```

12.1.7. Authentication

When authentication is enabled,

- The maximum number of advertised prefixes is reduced to 24.
- The first route entry in each RIPv2 message would be carrying 20 bytes of authentication data.
- If cryptographic authentication methods are used, further authentication data is placed after the entire RIPv2 message.

Task: Enable RIP authentication

```
(config-if)# ip rip authentication key-chain <name>
(config-if)# ip rip authentication mode {text | md5}
```

Use *show key chain" to spot invisible blank space after passwords

12.1.8. Summarization

- Default: auto-summarization
  - Summarizes prefixes to the classful network boundaries when classful network boundaries are crossed.

- Supernet advertisement not allowed
  - E.g. **ip summary-address rip 10.0.0.0 252.0.0.0**

Task: Disable automatic route summarization

```
(config-router)# no auto-summary
```

Task: Summarize a prefix

```
(config-if)# ip summary-address rip <ip-address> <mask>
```

12.1.9. Route updates

Task: Disable sending RIP updates on an interface but continue to receive the update

```
(config-if)# passive-interface { default | <type number>}
```

Task: Disable the validation of the source IP address of incoming RIP routing updates

```
(config-router)# no validate-update-source
```

Task: Send updates as broadcast

```
(config-if)# ip rip v2-broadcast
```

Task: Send updates as unicast

```
(config-router)# neighbor <ip-address>
```

12.1.10. Route filtering

Task: Stop advertising a route with a prefix-list

```
(config-router)# distribute-list prefix-list <name> {in | out}
```

Task: filter out RIP routes with extended access lists

```
(config-router)# distribute-list <extended-acl> {in|out} [<interface-id>]
```

- The source field in the ACL matches the update source of the route
- The destination field represents the network address

12.1.11. Route metric

- 16 unreachable network
- Ripv2 adds 1 to the route metric while sending updates.
  - RIPNg and EIGRIP increment metric when they receive updates
- maximum routes with same metric to the same subnet
  - 4 by default

Task: Add an offset to incoming and outgoing metrics to RIP routes

```
(config-router)# offset-list {<acl>} {in | out } <offset> {interface-type-number>}
```

12.1.12. Split horizon

Task: Disable split horizon

```
(config-if)# no ip split-horizon
```

12.1.13. Interpacket delay for RIP updates

- Useful when high-end router send RIP updates to low-end router
- Default: 0 in range 8 to 50 milliseconds

Task: Configure interpacket delay

```
(config-if)# output-delay <milliseconds>
```

12.1.14. Rip Optimization over WAN

Task: Enable triggered extensions for RIP

```
(config)# int serial <controller-number>
(config-if)# ip rip triggerred
```

12.1.15. Timers

Task: Configure RIP timers

```
(config-router)# timers basic <update> <invalid> <holddown> <flush> [<sleeptime>]
```

Update timer

Interval between updates.

- Default: 30 seconds

Invalid After timer
: time in seconds after which a route is declared invalid.

- Default: 180 seconds
- Reset after update is received
- Should be at least 3 times the update timer.
- Invalid routes are still used for forwarding packets

Holdown timer
: interval during which routing information about better paths is suppressed.

- Default: 180 seconds
- Should be at least 3 times the update timer
- The route is marked inaccessible and advertised as unreachable.
- Holdown routes are still used for forwarding packets

Flush After timer
: amount of time that must pass before a route is removed from the RIB.

- Default: 240 seconds
- Starts at the same time than Invalid After timer
- Cisco IOS checks this timer only after the Invalid After timer expired
  - No consequence If Flush timer < Invalid Timer

Sleep time
: amount of time for which routing updates will be postponed.

Task: Specify a default update interval on an interface

```
(config-if)# ip rip advertise <seconds>
```

- The command above overrides the update timers set by **timers basic** command.

12.2. EIGRP

- classless protocol (VLSM, summarization)
- multiple routed protocol support (ipv4, ipx, appletalk,)
- uses its own transport protocol
  - IP protocol 88: RTP
  - Uses multicast to 224.0.0.10 and unicast

- Administrative distance : 90 internal routes, 5 summary routes , 170 external routes
- Forms active neighbor adjacencies
- DUAL for loop-free topology and fast convergence
- granular metric
- unequal cost load balancing
- summarization
- Supports MD5 based authentication

12.2.1. EIGRP Packet Format

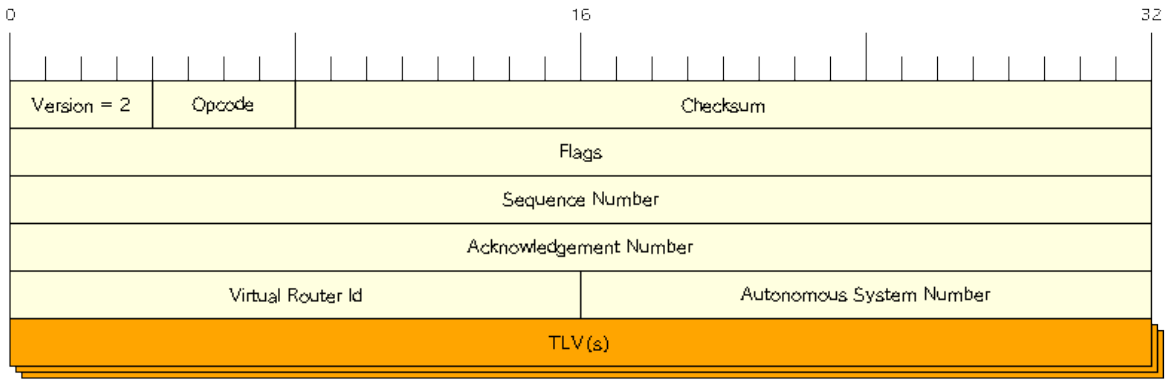


Figure 18. EIGRP packet format

Opcode

EIGRP packet type

- 1 = Update, 3 = Query, 4 = Reply, 5 = Hello/Ack, 10 = SIA Query, 11 = SIA Reply.
- Other types have been allocated for different, mostly unimplemented purposes, or are obsolete

Checksum

- based on the entire EIGRP packet excluding the IP header.

Flags

- 0x1 = Init (used during initial adjacency buildup)
- 0x2 = Conditional Receive (used by RTP to allow this message to be received only by a subset of receivers)
- 0x4 = Restart (indicates that a router has restarted)
- 0x8 = End-of-Table (indicates that the transmission of the entire EIGRP database is complete).

Sequence

- Facilates orderly delivery of packets

Acknowledgement

- sequence number of the last packet heard from the neighbor to which this packet is being sent.
- A Hello packet with a nonzero ACK field will be treated as an ACK packet rather than as a Hello.
- only unicast because acknowledgments are never multicasted.

TLV

- Field used to carry route entries as well as provide EIGRP DUAL information.
  - 0x0001 EIGRP Parameters (General TLV Types)
  - 0x0002 Authentication Type (General TLV Types)
  - 0x0003 Sequence (General TLV Types)
  - 0x0004 Software Version (General TLV Types)
  - 0x0005 Next Multicast Sequence (General TLV Types)
  - 0x0102 IPv4 Internal Routes (IP*Specific TLV Types)
  - 0x0103 IPv4 External Routes (IP*Specific TLV Types)
  - 0x0402 IPv6 Internal Routes (IP*Specific TLV Types)
  - 0x0403 IPv6 External Routes (IP*Specific TLV Types)
  - 0x0602 Multi Protocol Internal Routes (AFI*Specific TLV Types)
  - 0x0603 Multi Protocol External Routes (AFI*Specific TLV Types)

12.2.2. EIGRP messages

Task: Show Statistics about messages sent and received

```
# show ip eigrp traffic

EIGRP-IPv4 VR(CCIE) Address-Family Traffic Statistics for AS(1)
  Hellos sent/received: 1132/6090
  Updates sent/received: 169/428
  Queries sent/received: 0/0
  Replies sent/received: 0/0
  Acks sent/received: 74/191
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 246
  PDM Process ID: 244
  Socket Queue: 0/10000/7/0 (current/max/highest/drops)
  Input Queue: 0/2000/7/0 (current/max/highest/drops)
```

Hello

- Opcode = 5
- Multicast to 224.0.0.10

- Do not require acknowlegment
- Can be used as Ack if sent without data
- every 5 seconds or 60 seconds on NBMA interfaces with < 1 Mbps bandwidth
- Non-reliable

Ack

- unicast in response to Update, Query, Reply, SIA-Query, and SIA-Reply packets
- contains a nonzero acknowledgement number set to the Sequence number of the reliable packet being acknowledged.
- uses the same Opcode as the Hello packet
- Non-reliable

it is allowed to use any unicast reliable packet to also carry an acknowledgment number. If a router has both a unicast reliable packet to send to a neighbor and also needs to acknowledge a previously received reliable packet from that neighbor, the sequence number of the received reliable packet can be sent along with the outbound reliable packet in its Acknowledgment number field. It is not necessary to send a standalone ACK in this case; the unicast reliable packet carrying a nonzero Acknowledgment number field will be processed by its recipient both by its true type and as an ACK.

Update

- multicast or unicast
  - unicast during a new adjacency buildup, Update packets are unicasted between the newly discovered neighbors.
    - In specific cases, when multiple new neighbors are detected on a single multiaccess interface in a short time span, EIGRP might choose to synchronize to them using multicasts for efficiency reasons (for example, when a hub router in a DMVPN network starts and detects tens or hundreds of spoke routers).
  - multicast after routers have fully synchronized
  - unicast if a neighbor does not acknowledge the arrival of an Update packet
  - always unicasts on point-to-point interfaces and for statically configured neighbors

Query

- Opcode = 3
- multicast unless in response to a received query

Reply

- Opcode = 4
- unicast
- indicates that it does not need to go into Active state because it has a FS

Request

- unicast or multicast
- get specific info from neighbors
- used in route server applications

SIA-Query

- Opcode = 10
- unicast
- used during a prolonged diffusing computation to verify whether a neighbor that has not yet sent a Reply to a Query is truly reachable and still engaged in the corresponding diffusing computation. The SIA-Query packet is used to ask a particular neighbor to confirm that it is still working on the original Query. If the neighbor is reachable and is still engaged in the diffusing computation for the destination specified in the SIA-Query, it will immediately respond with an SIA-Reply packet. As a result, the timer that governs the maximum time a diffusing computation is allowed to run is reset, giving the computation extra time to finish

SIA-Request

- Opcode = 10
- unicast
- Unreliable packets: Hello and Ack
- Reliable packets: Update, Query/Reply, SIA-Query/SIA-Reply
  - Must be ACK
  - are retransmitted at most 16 times

Task: debug EIGRP

```
debug ip eigrp packet [hello | ack | update } quey | reply]
```

12.2.3. Neighbors

- Discovered with Hello packets
- can be set manually
- must agree on
  - Primary IPv4 subnet
  - Autonomous System Number
  - Authentication
  - K values
- Do not need to agree on timers
  - The hold time is included in the hello packets so each neighbor should stay alive even though the hello interval and hold timers do not match.

After a static neighbor is defined, all EIGRP multicasts on the interface through which the neighbor is reachable will be disabled. As a result, EIGRP-enabled routers will not establish an adjacency if one router is configured to use unicast (static) while another uses multicast (dynamic) on the same link. Here's another way of putting this rule: Either all neighbors on a common network segment are statically configured for each other, or none of them are.

Task: Adjust EIGRP Hello interval

```
(config-if)# ip hello-interval eigrp <asn> <seconds>
```

Task: Adjust EIGRP Holdown time

```
(config-if)# ip hold-time eigrp <asn> <seconds>
```

Changing the Hello interval does not result in automatic recalculation of the Hold time. This can, under certain circumstances, result in problems with flapping adjacencies if the Hello interval is manually configured to be close or even higher than the default Hold time, without changing the Hold timer itself.

Task: Verify neighbor adjacencies

```
# sh ip eigrp neighbors [detail]

IP-EIGRP neighbors for process 1
H  Address    Interface  Hold Uptime    SRTT   RTO  Q  Seq
                         (sec)          (ms)       Cnt Num
1  10.10.10.3 Fa0/0       11   00:00:08    87   522  0   6
0  10.10.10.2 Fa0/0       14   00:01:54  1300  5000  0   3
```

Q Cnt indicates the number of enqueued reliable packets, that is, packets that have been prepared for sending and even possibly sent but for which no ACK has been received yet from the neighbor. In a stable network, the Q Cnt value must be zero; non- zero values are normal during initial router database synchronization or during network convergence. If the Q Cnt value remains nonzero for prolonged periods of time, however, it indicates a communication problem with the neighbor.

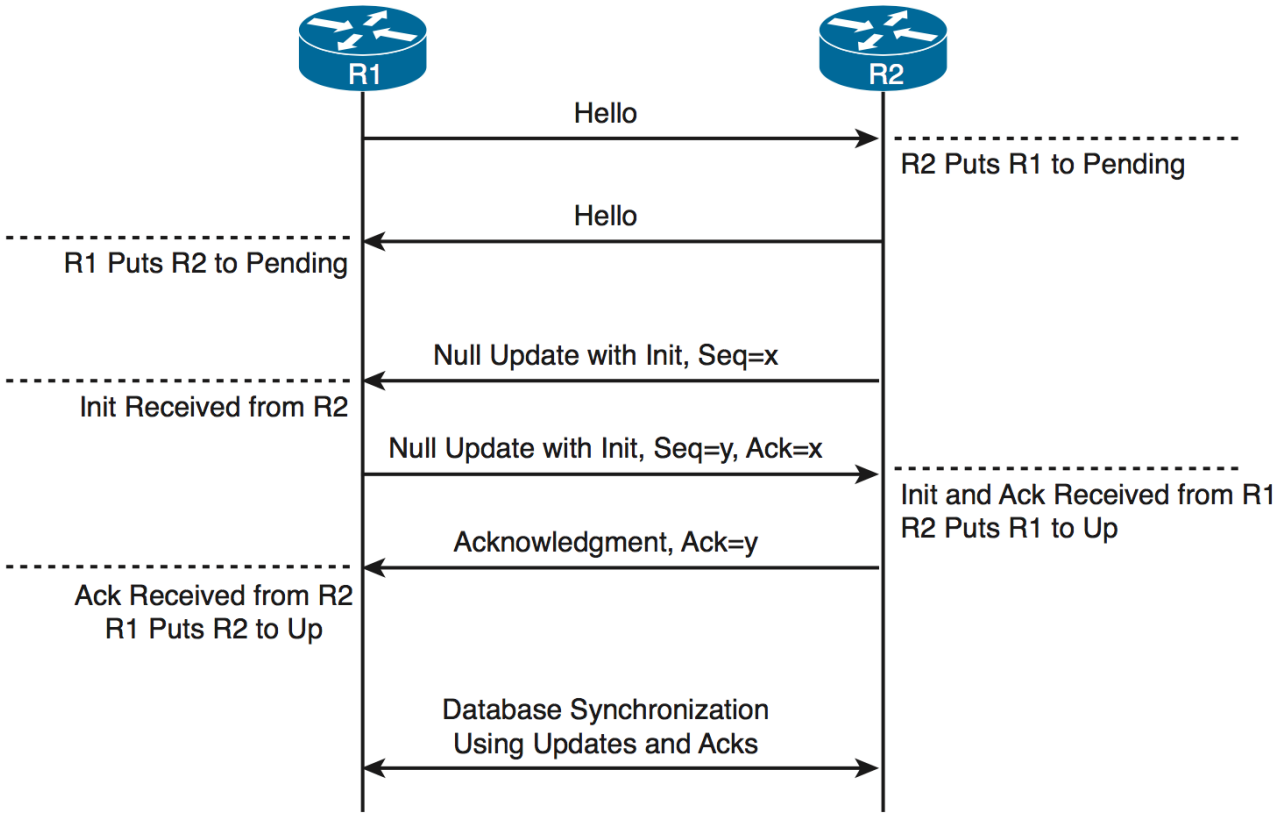
Task: Exchange EIGRP packets only as unicast

```
(config-router)# neighbor <a.b.c.d> <interface-id>
```

Task: Exchange EIGRP packets only as unicast in named configuration

```
(config-router-af-interface)# neighbor <a.b.c.d> <interface-id>
```

Adjancency creation



:EIGRP does not build peer relationships over secondary addresses. All EIGRP traffic is sourced from the primary address of the interface.

12.2.4. EIGRP Loop prevention techniques

Split horizon

- Enabled by default on all interfaces

Task: Disable split horizon for EIGRP

```
(config-if)# no ip split-horizon eigrp <asn>
```

Task: Disable split horizon in named configuration

```
(config-router-af-interface)# no split-horizon
```

12.2.5. Metric

Metric = 256 * ((K_1 * B\a\n\dwidth + (K_2 * B\a\n\dwidth)/(256 - Load) + K_3 * Delay)) * (K_5 / (Reliability + K_4))

- Default Values: k1,k2,k3,k4,k5 = 1,0,1,0,0
- The values of K must match for the neighbors to become adjacents

- EIGRP uses integer division while calculating the metric

Task: Description

```
(config-router)# metric weights
```

Bandwith Metric Component

- frac {10^(7)} { "minimum Bandwidth in Kbps"}
- Range: 1 Mbps to 10 Gpbs

Task: Configure the bandwidth of an interface

```
(config-if)# bandwidth <kbps>
```

Delay Metric Component

- in tens-of-microseconds
- sum of delay on the path to the destination
- Range: 1..167,772,14
- EIGRP split horizon with poison reverse, route withdrawal uses max delay 167,772,15 to indicate an unreachable network
- CAUTION: **show ip interface** displays delay in micro-seconds

Task: Configure the delay of an interface

```
(config-if)# delay <tens-of-microseconds>
```

Reliability Metric Component

- likelihood of successful packet transmission with 0 means 0% and 255 means 100%
- Minimum value along the path
- EIGRP does not send a new update every time the reliability changes along the path
- The reliability metric of a route is just a snapshot of its then-current reliability when it was last advertised.

Load Metric Component

- Maximum effective Txload of the route with 255 means 100% loading
- To account for large differences in the momentary load caused by bursty traffic, IOS actually computes an exponentially weighted average over the momentary load that smooths out short-lived load swings.
- Because an interface can be differently utilized in the ingress and egress data flow direction, IOS maintains two independent load metric counters, the Txload for outgoing traffic and Rxload

for incoming traffic.

- EIGRP does not send a new update every time the load changes along the path
- The load metric of a route is just a snapshot of its then-current load when it was last advertised.

MTU Metric Component

- minimum Maximum transmission unit
- not factored into the composite metric calculation and does not impact the best-path selection in any way

Hop Count Metric Component

- Default max value: 100, can be set to 255
- not factored into the composite metric calculation and does not impact the best-path selection in any way

Routing Metric Offset Lists

TODO

When trying to manually influence EIGRP path selection through interface bandwidth/delay configuration, the modification of bandwidth is discouraged for following reasons:

- The change will only affect the path selection if the configured value is the low- est bandwidth over the entire path. Changing the bandwidth can have impact beyond affecting the EIGRP metrics. For example, QoS also looks at the bandwidth on an interface.
- EIGRP by default throttles to use 50 percent of the configured bandwidth. Lowering the bandwidth can cause problems like starving EIGRP neighbors from getting packets because of the throttling back. Configuring an excessively high bandwidth can lead EIGRP to consume more bandwidth than physically avail- able, leading to packet drops.
- Changing the delay does not impact other protocols nor does it cause EIGRP to throttle back, and because, as it's the sum of all delays, has a direct effect on path selection.

12.2.6. Wide Metric

Metric = [(K1*Minimum Throughput + (K2*Minimum Throughput/(256-Load) + (K3*Total Latency) + (K6*Extended Attributes)]* [K5/(K4 + Reliability)]

- Use one of the following commands to confirm wide metric support:
  - **sh eigrp plugins**
  - **sh eigrp tech-support**
  - **sh ip protocols**

TODO: .Task: Change the scale

```
(config-router)# metric rib-scale <1..255>
```

- throughput → bandwidth

Latency Metric Component

- ~ delay
- On interfaces physically operating on speeds of 1 Gbps and lower without bandwidth and delay commands, the interface delay is simply its IOS-based default delay converted to picoseconds.
- On interfaces physically operating on speeds over 1 Gbps without bandwidth and delay commands, the interface delay is computed as 10^{13} / interface default bandwidth.
- On interfaces configured with the explicit bandwidth command and without the delay command, regardless of their physical operating speed, the interface delay is the IOS-based default delay converted to picoseconds.
- On interfaces configured with explicit delay command, regardless of their physical operating speed and the bandwidth setting, the interface delay is computed as its specified delay value converted to picoseconds, that is, 10^{7} * value of the delay command (recall that the delay command defines the delay in tens of microseconds)

12.2.7. Reliable Transport Protocol

- guarantees delivery in order
- Update, Query, Reply, SIA-Query, SIA-Request packets
- uses Conditional Receive for reliable and efficient multicast
  - partition all its neighbors on a multiaccess interface into two groups: a group of well-behaved neighbors that have been able to acknowledge all multicast messages sent so far and a group of “lagging” routers that have failed to acknowledge at least one transmitted reliable EIGRP packet and that must be handled individually. If EIGRP wants to continue sending the multicast packets in parallel with retransmitting the unacknowledged packets to the lagging routers as unicasts, it has to send the in-order multicast packets with a special flag saying “this packet is only for those routers that have received all multicast packets so far.”
  - accomplished by the sender first transmitting a Hello packet with two specific TLVs called the Sequence TLV and the Next Multicast Sequence TLV, often called a Sequenced Hello. The Next Multicast Sequence TLV contains the upcoming sequence number of the next reliable multicasted message. The Sequence TLV contains a list of all lagging neighbors by their IP address, in effect saying “whoever finds himself in this list, ignore the next multicast message with the indicated sequence number.” A neighbor receiving this Sequenced Hello packet and not finding itself in the Sequence TLV will know that it is expected to receive the upcoming multicast packet, and will put itself into a so-called Conditional Receive mode (CR-mode). A neighbor receiving this Sequenced Hello packet and finding itself in the Sequence TLV, or a neighbor not receiving this Hello packet at all for whatever reason will not put

itself into the CR-mode. Afterward, the sending router will send the next multicast packet with the CR flag set in its Flags field. Routers in CR-mode will process this packet as usual and then exit the CR-mode; routers not in CR-mode will ignore it. As a result, the router is able to continue using multicast with those routers that have no issues receiving and acknowledging it, while making sure that the lagging neighbors won't process the multicasts until they are able to catch up. Each lagging neighbor that has not acknowledged one or more multicast packets will be sent these packets as unicasts in their proper sequence.

- multicast flow timer: time to wait for an ACK before declaring a neighbor as lagging and switching from multicast to unicast
- RTO (Retransmission Time Out): the time between the subsequent unicasts
- SRTT (Smooth Round Trip Time): is average elapsed time, measured in milliseconds, between the transmission of a reliable packet to the neighbor and the receipt of an acknowledgment.

12.2.8. EIGRP Autonomous System Configuration

- created with the command **router eigrp** <autonomous-system-number>
- EIGRP VPNs can be configured only under IPv4 address family. A VRF instance and route distinguisher must be defined before the address family session can be created.
- recommendation: configure the asn when the address family is configured by **router eigrp** <asn> **address-family** or seperately using the **autonomous-system** command.

12.2.9. EIGRP Named Configuration

- Global params under SAFI or in **config-router-topology base** mode
- interface params in **config-router-af-interface** mode
- wide-meric scaling automatic enabled
- can be configured in IPv4 and IPv6 named configuration
- VRF instance and a RD are optional
- EIGRP IPv6 VRF-lite feature is available only in EIGRP named configuration
- EIGRP VPNs can be configured. A VRF and RD must be defind before the address-family session can be created.
- a single EIGRP routing process can support multiple VRFs. However, a single VRF can be supported by each VPN . Redistribution between VRFs is not supported.

Task: Configure a basic EIGRP named configuration

```
(config)# router eigrp <virtual-instance-name>
(config-router)# address-family ipv4 [multicast] [umicast] [vrf <vrf-name>]
autonomous-system <asn>
(config-router-af)# network <a.b.c.d>
```

Task: Convert Classic Configuration to EIGRP named configuration

```
# eigrp upgrade-cli name
```

Address Family Section

```
(config-router-af)# ?
Address Family configuration commands:
  af-interface        : Enter Address Family interface configuration
  default             : Set a command to its defaults
  eigrp               : EIGRP Address Family specific commands
  exit-address-family : Exit Address Family configuration mode
  maximum-prefix      : Maximum number of prefixes acceptable in aggregate
  metric              : Modify metrics and parameters for advertisement
  neighbor            : Specify an IPv4 neighbor router
  network             : Enable routing on an IP network
  shutdown            : Shutdown address family
  timers              : Adjust peering based timers
  topology            : Topology configuration mode
```

Per-AF-Interface Section

```
(config-router-af-interface)# ?
Address Family Interfaces configuration commands:

add-paths          : Advertise add paths
authentication     : authentication subcommands
bandwidth-percent  : Set percentage of bandwidth percentage limit
bfd                : Enable Bidirectional Forwarding Detection
dampening-change   : Percent interface metric must change to cause update
dampening-interval : Time in seconds to check interface metrics
default            : Set a command to its defaults
exit-af-interface  : Exit from Address Family Interface configuration mode
hello-interval     : Configures hello interval
hold-time          : Configures hold time
next-hop-self      : Configures EIGRP next-hop-self
passive-interface  : Suppress address updates on an interface
shutdown           : Disable Address-Family on interface
split-horizon      : Perform split horizon
summary-address    : Perform address summarization
```

Per-AF-Topology Configuration Section

Within the context of Multi Topology Routing, a topology is defined as a subset of rout- ers and links in a network for which a separate set of routes is calculated. The entire net- work itself, for which the usual set of routes is calculated, is known as the base topology. The base topology is the default routing environment that exists prior to enabling MTR. Any additional topologies are known as

class-specific topologies and are a subset of the base topology. Each class-specific topology carries a class of traffic and is characterized by an independent set of Network Layer Reachability Information (NLRI) that is used to maintain separate routing tables and FIB databases. This design allows the router to per- form independent route calculation and forwarding for each topology. Multiple topolo- gies can be used to segregate different classes of traffic, such as data, voice, and video, and carry them over different links in the same physical network, or to keep separate and independent topologies for IPv4 and IPv6 routing. Multiple topologies are not equivalent to Virtual Routing and Forwarding (VRF) tables because they share the common address space, and they are not intended to provide address conservation or reuse.

EIGRP is capable of keeping separate routing information for different topologies, and its behavior per specific topology within an address family can be configured in the per-AF- topology section. On routers without MTR support, only the topology base command will be available; on routers supporting MTR, the topology command will allow referenc- ing a particular separate topology table definition by its name.

```
(config-router-af-topology)# ?
Address Family Topology configuration commands:

auto-summary        : Enable automatic network number summarization
default             : Set a command to its defaults
default-information : Control distribution of default information
default-metric      : Set metric of redistributed routes
distance            : Define an administrative distance
distribute-list     : Filter entries in eigrp updates
eigrp               : EIGRP specific commands
exit-af-topology    : Exit from Address Family Topology configuration mode
maximum-paths       : Forward packets over multiple paths
metric              : Modify metrics and parameters for advertisement
offset-list         : Add or subtract offset from EIGRP metrics
redistribute        : Redistribute IPv4 routes from another routing protocol
snmp                : Modify snmp parameters
summary-metric      : Specify summary to apply metric/filtering
timers              : Adjust topology specific timers
traffic-share       : How to compute traffic share over alternate paths
variance            : Control load balancing variance
```

12.2.10. DUAL

Topology Table

Task: Display EIGRP topology table

```
# show ip eigrp topology [as-number | [[ip-address] mask]] [active | all-links |
pending | summary | zero-successors]

IP-EIGRP Topology Table for process 77

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
      r - Reply status

P 172.16.90.0 255.255.255.0, 2 successors, FD is 0
         via 172.16.80.28 (46251776/46226176), Ethernet0
         via 172.16.81.28 (46251776/46226176), Ethernet1
         via 172.16.80.31 (46277376/46251776), Serial0
P 172.16.81.0 255.255.255.0, 1 successors, FD is 307200
         via Connected, Ethernet1
         via 172.16.81.28 (307200/281600), Ethernet1
         via 172.16.80.28 (307200/281600), Ethernet0
         via 172.16.80.31 (332800/307200), Serial0
```

P - Passive

No EIGRP computations are being performed for this destination.

A - Active

EIGRP computations are being performed for this destination.

U - Update

Indicates that an update packet was sent to this destination.

Q - Query

Indicates that a query packet was sent to this destination.

R - Reply

Indicates that a reply packet was sent to this destination.

r - Reply

status Flag that is set after the software has sent a query and is waiting for a reply.

RD

Reported Distance

CD

Computed Distance

FD

Feasible Distance

- record of the lowest known distance since the last transition from the Active to Passive state.

    - In other words, FD is a historical record, or a historical copy, of the smallest known CD toward a particular destination, with the history starting anew with the last Active-to-Passive transition.
    - Being a record of the smallest known CD since the route entered the Passive state for the last time, FD is not necessarily equal to the current best CD to a destination.
    - By its definition, in the Passive state, after the FD has been initialized, it can only decrease (if the current best CD happens to fall below the current value of FD) or remain at its current value (if the current best CD rises but the route remains Passive).
    - There is exactly one FD per each destination, regardless of the number of neighbors.
    - FD is an internal variable maintained for each network known to EIGRP whose value is never advertised to another router.
  - lowest bandwidth on the path to this destination as reported by the upstream neighbor
  - total delay
  - path reliability
  - path loading
  - minimum path maximum transmission unit (MTU)
  - feasible distance
  - reported distance
  - route source (external routes are marked)

Feasibility condition

- Feasibility condition: RD < FD
  - it is a sufficient condition but not a necessary condition
  - not every loop-free path satisfies the FC
  - proven by Dr. J. J. Garcia-Luna-Aceves
  - also called the Source Node Condition
- Feasible Successor: Neighbor that satisfy the FC
- successor: Feasible Successor with the least CD

Topology Changes

- A topology change occurs whenever the distance to a network changes or a new neighbor comes online that advertises the network.
  - The distance change can be detected either through receiving an Update, Query, Reply, SIA-Query, or SIA-Reply packet from a neighbor that carries updated metric information about the network, or because a local interface metric has changed.
  - Also, the event of a neighbor going down is processed by setting the CD/RD of all networks reachable through that neighbor to infinity.

- Whenever EIGRP detects a topology change,
  - it first records the change into the topology table and updates the RD and CD of the neighbor that advertised the change (in case of a received EIGRP message) or was influenced by it (in case of a link metric change).
  - From among all neighbors that advertise the network, EIGRP identifies the one that provides the least CD, taking into account the updated CDs. Note that the FC is not invoked at this step.
- Only after identifying the neighbor offering the least CD, EIGRP verifies whether this neighbor meets the FC and is therefore a Feasible Successor. If it is, EIGRP will promote it to the Successor and start using it right away. If, however, that neighbor does not meet the FC, EIGRP will put the route into the Active state and send out Queries, asking its neighbors to assist in locating the best route.

Local Computation

- After a topology changes, if the best path is through a Feasible Successor, do the following:
  1. The Feasible Successor providing the least CD is made the new Successor.
  2. If the CD over the new Successor is less than the current FD, the FD will be updated to the new CD; otherwise it stays at its current value.
  3. The routing table is updated to point toward the new Successor.
  4. If the current distance to the destination has changed as a result of switching to a new Successor, an Update packet is sent to all neighbors, advertising the router's updated distance to the destination.

Diffusing Computation

If after a topology changes , if the router finds out that the new shortest path is provided by a neighbor that is not a Feasible Successor, do the following:

1. The entry in the routing table, still pointing to the current unchanged Successor, is locked: It must not be removed nor its next hop changed until the diffusing compu- tation is finished and the route has been moved to the Passive state again.
2. The FD is set to the current (possibly increased) CD through the current unchanged Successor. Also, if this router ever needs to advertise its distance to the network while in the Active state, it will also use the value of the current CD through the Successor.
3. The network is put into the Active state and the router sends out a Query packet to all its neighbors. This Query packet contains the Active network's prefix and the router's current CD toward it.

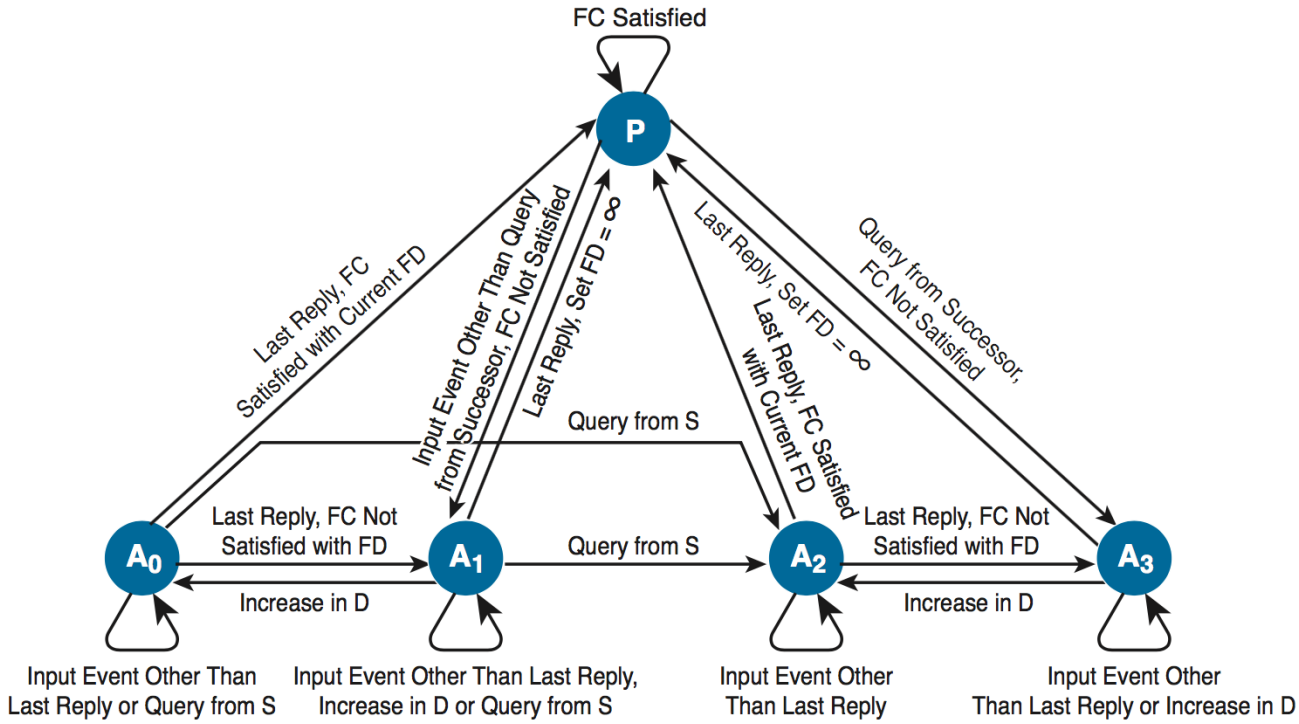
One single topology change

Each neighbor receiving a Query packet will process it by updating its own topology table using the distance information advertised in the Query and reevaluating its own choice of Successors and Feasible Successors. Two possibilities now exist: Either the neighbor still has its own Feasible Successor or a Successor that provides it with the least- cost loop-free path, or the information contained in the Query causes the neighbor to stop considering the path through its current

Successor the shortest available and none of its own neighbors that offer the shortest path are a Feasible Successor.

Multiple topology changes

- Uses DUAL Finite State Machine to handle multiple topology changes occuring a simple diffusing computation



States

- P : Passive
- A0: Local Origin with Distance Increase
- A1: Local Origin
- A2: Multiple Origins
- A3: Successor Origin

Rules

- Unless a change in distance occurs such that the neighbor providing the least CD fails to meet the FC, the route remains passive.
- If a Query is received from the current Successor and, after processing the distance indicated in this Query, the neighbor that provides the least CD fails to meet the FC, the route will enter the A3 active state.
  - The router will send out Queries and wait for Replies.
  - If no further distance increase is detected while waiting for the Replies, the last Reply allows the router to transition back to the Passive state, reinitialize the FD, and choose any neighbor that provides the least CD as the new Successor.
- If a distance change caused by other means than a Query from a Successor is detected (this

can be caused by receiving an Update, changing an interface metric, or losing a neighbor) and after processing the change, the neighbor that provides the least Computed Distance fails to meet the Feasibility Condition, the route will enter the A1 active state, also called the Local Origin Active State. The router will send out Queries and wait for Replies. If no further distance increase or Query from the current Successor is received while waiting for the Replies, the last Reply allows the router to transition back to the Passive state, reinitialize the Feasible Distance, and choose any neighbor that provides the least Computed Distance as the new Successor.

- If during the stay in the A3 (Successor Origin) or A1 (Local Origin) active states, another distance increase caused by other means than the Successor's Query is detected, another topology change during the diffusing computation has occurred. Because the router cannot advertise this updated distance while it is in the Active state, other routers might not be informed about it and their Replies might not take this new increased distance into account. Therefore, extra scrutiny is applied to the received Replies instead of simply choosing the neighbor that provides the least Computed Distance. This is accomplished first by changing the state from A3 (Successor Origin) to A2 (called Multiple Origins), or from A1 (Local Origin) to A0 (no official name; we will call it Local Origin with Distance Increase) states. In A2 or A0 states, the router waits to receive all remaining Replies. When the last Reply arrives, the router will first check whether the neighbor providing the least Computed Distance passes the Feasible Condition check using the Feasibility Distance value set when the route entered the Active state (recall that it was set to the increased distance through the current Successor at the moment of transition- ing to the Active state). This extra check essentially mimics a situation in which the router is actually using the path through the current Successor and has just detected the distance increase, so it uses the current value of Feasibility Distance to verify whether the neighbor providing the least Computed Distance passes the Feasibility Condition. If it does, the route becomes Passive again, and the neighbor is chosen as the Successor. If it does not, however, the route will return from A0 (Local Origin with Distance Increase) to A1 (Local Origin) or from A2 (Multiple Origins) to A3 (Successor Origin) and the router will commence another diffusing computation by again sending a Query.
- If during the stay in A1 (Local Origin) or A0 (Local Origin with Distance Increase) active states a Query from the Successor is received, another topology change dur- ing the diffusing computation has occurred. Because the router cannot advertise this updated distance while it is in the Active state, other routers might not be informed about it and their Replies might not take this new increased distance into account. Therefore, extra scrutiny is applied to the received Replies. This is accomplished by changing the state to A2 (Multiple Origins) and then proceeding from that state just like in the previous case

Task: Display details on EIGRP Active States

```
# sh ip eigrp topology active
```

Stuck-In-Active

- when all expected Replies are not received before the **Active** timer (default= 3 minutes) expires after first Query
  - The neighbors that did not reply will be removed from the neighbor table and their

adjacencies torn down, and the diffusing computation will consider these neighbors to have responded with an infinite metric.

- If a neighbor does not respond to a Query message with its Reply within half of the Active timer time, the router will send the neighbor a SIA-Query message. The SIA- Query stands for a message saying “Are you still working on my Query?” If the neigh- bor is able to receive and process this SIA-Query, it will immediately respond with the SIA-Reply message. The contents of the SIA-Reply can either say “Yes, I still expect my own neighbors to send me the Replies I’ve asked them for” or “No, the computation is finished; this is my current metric to the destination.” In any case, the SIA-Reply is sent immediately as a response to the SIA-Query message; there is nothing to wait for. Receiving an SIA-Reply allows the Active timer to be reset, giving the diffusing computa- tion an additional time to complete. At most three SIA-Queries can be sent, each after half of the Active timer. If the diffusing computation is not finished by the time the third SIA-Query was replied to by an SIA-Reply and the half of the Active timer expired again, the adjacency to the neighbor will be dropped. The same will happen if an SIA-Query is not responded to by an SIA-Reply within the next half of the Active timer. With the default setting of the Active timer to 180 seconds, three consecutive SIA-Query packets allow extending the diffusing computation to a maximum of 4 × 90 = 360 seconds (90 seconds to the first SIA-Query, plus each SIA-Query buying another 90 seconds).

Task: Control the time that the router waits (after sending a query) before declaring the route to be in the stuck in active state.

```
(config-router)# timers active-time [<minutes>| disabled]
```

default wait time = 3 minutes

- Reasons a router doesn’t respond to EIGRP Query:
  - The neighbor router’s CPU is overloaded and the router either cannot respond in time or is even unable to process all incoming packets including the EIGRP packets.
  - Quality issues on the link are causing packets to be lost.
  - Low-bandwidth links are congested and packets are being delayed or dropped.
  - The network topology is excessively large or complex, either requiring the Query to propagate to a significant depth or causing an inordinate number of prefixes to be impacted by a single link or node failure.
- Troubleshooting SIA routes is generally a three-step process:
  1. Find the routes that are consistently being reported as SIA.
  2. Find the router that is consistently failing to answer queries for these routes
  3. Find the reason that router is not receiving or answering queries.

The first step should be fairly easy. If you are logging console messages, a quick perusal of the log indicates which routes are most frequently marked SIA.

The second step is more difficult. The command to gather this information is show ip eigrp topology

active:

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

A 10.2.4.0/24, 0 successors, FD is 512640000, Q
    1 replies, active 00:00:01, query-origin: Local origin
         via 10.1.2.2 (Infinity/Infinity), Serial1
    1 replies, active 00:00:01, query-origin: Local origin
         via 10.1.3.2 (Infinity/Infinity), r, Serial3
    Remaining replies:
         via 10.1.1.2, r, Serial0
```

Any neighbors that show an R have yet to reply (the active timer shows how long the route has been active). Note that these neighbors may not show up in the Remaining replies section; they may appear among the other RDBs. Pay particular attention to routes that have outstanding replies and have been active for some time, generally two to three minutes. Run this command several times and you begin to see which neighbors are not responding to queries (or which interfaces seem to have a lot of unanswered queries). Examine this neighbor to see if it is consistently waiting for replies from any of its neighbors. Repeat this process until you find the router that is consistently not answering queries. You can look for problems on the link to this neighbor, memory or CPU utilization, or other problems with this neighbor.

If you run into a situation where it seems that the query range is the problem, it is always best to reduce the query range rather than increasing the SIA timer.

12.2.11. Stub Routing

TODO Better explanation of this feature

- improves network scalability and stability.
- commonly used in hub-and-spoke networks.
- configured only on spoke routers.
- announces its stub router status using an additional TLV in its EIGRP Hello messages.

The results of configuring a router as a stub are multifold:

- A stub router does not propagate routes learned through EIGRP to its neighbors, with the exception of **leak-map** routes . This prevents a stub router from ever being considered a Feasible Successor for remote networks by its neighbors and possibly becoming a transit router at some point in the future.
- A stub router advertises only a subset of its own EIGRP-enabled networks to its neighbors. This subset can be defined in the **eigrp stub** command using the **summary**, **connected**, **static**, **redistributed**, and **receive-only** keywords.
- Neighbors of a stub router aware of its stub status (thanks to the specific TLV in the stub router's Hello packets) will never send a Query packet to a stub router. This prevents the neighbors from converging through a stub router to reach networks that are remote to the stub

router.

The following rules summarize the stub router behavior with respect to handling Query packets:

- Originating Query packets is not modified in any way. Rules for entering the Active state and sending Queries are precisely the same.
- Processing received Query packets depends on what network was queried for. If the network in the received Query is a network the stub router is allowed to adver- tise, meaning that it falls under the configured category of summary, connected, static, or redistributed, the router will process the Query normally (even possi- bly causing the stub router to become Active itself) and send back an appropriate Reply. The same is valid for an EIGRP-learned network that is allowed to be further advertised using a leak-map—a Query for such a network would be processed and responded to in the usual way. If the Query contains a network that the stub router knows about but is not allowed to advertise (the network does not fall under the configured category, or is learned through EIGRP but not allowed for further adver- tisement by a leak-map), it will be processed in the usual way as described earlier, but the Reply will always indicate infinite distance, regardless of what the stub router truly knows about the network. Receiving a Query for an unknown network will immediately cause the router to respond with a Reply and an infinite distance; how- ever, this is regular EIGRP behavior not related to the stub feature.
- At this point, you might ask why a stub router would receive a Query, as its stub status should instruct its neighbors to avoid sending Queries to it. There are two primary rea- sons why even a stub router might receive a Query. First, a stub router's neighbor might be running an old IOS that does not recognize the stub TLV yet. Such a neighbor will cre- ate an adjacency to a stub router just fine, but it will also happily send Queries to it, not knowing that the router is a stub router. Second, if there are multiple routers on a com- mon segment and all of them are configured as stub routers, if any of these stub routers need to send a Query, it will also send it to all its stub neighbors. This is done to support multihomed branch offices that usually have two branch routers configured as stubs. Each of these branch routers is connected to the headquarters through its own uplink, and they are also connected together by a common intra-site link. If the uplink on one of the branch routers fails, the affected router needs to converge through its neighbor branch router, and this might require a permission to send Queries to its fellow stub neighbor. Therefore, on a common segment with all routers configured as stubs, Queries are sent as usual.
- In case of multiaccess segments with mixed neighbors (stub and nonstub), EIGRP solves the problem of sending Queries only to nonstub neighbors in two ways: Either it sends the Queries as unicasts to the nonstub neighbors or it uses the Conditional Receive mode in RTP to send multicast Queries in such a way that only nonstub routers will process them. The choice of a particular mechanism depends on the number of nonstub neighbors. While mixing stub and nonstub routers on a common segment is not a recom- mended practice, it is inevitable, for example, in cases where the hubs and spokes are interconnected by a DMVPN or a VPLS service.

Task: Configure EIGRP stub

```
(config-router)# eigrp stub {[received-only] | [connected] [static] [ leak-map <name>]
[redistributed] [summary]
```

receive-only

does not advertise any prefixes.

- only receives prefixes advertised to it by its neighbors.
- either static routing on its neighbors or NAT/PAT on the stub router is required in this case to allow the networks behind the stub router to communicate with the outside world.
- cannot be used with any other keywords when configuring stub routing.

leak-map

Allows some prefix to be advertised

- crucial in scenarios where a branch office uses a pair of interconnected routers configured as stub routers. If these routers are to provide backup connectivity to each other, they must be allowed to readvertise EIGRP-learned routes to each other, even in stub mode.

connected

Advertises connected subnets.

- directly connected interfaces will not be advertised automatically; it is still necessary to add them to EIGRP using the usual **network** command
- option enabled by default

static

Advertises static routes.

- The static routes need to be redistributed into EIGRP to be advertised.

summary

Advertises Summary routes

- summary routes can be created manually (**summary-address**) or automatically at a major network border router (**auto-summary**).
- option enabled by default

redistributed

Advertises redistributed routes

the stub router feature has no impact on what routes the hub router will advertise to its stub spokes. Without an additional configuration on the hub router, the spokes will be populated with full routing tables. Considering the fact that in a hub-and-spoke network, any other network beyond the branch networks is reachable through the hub, having full routing tables on spoke routers with most of their entries pointing toward the hub router is not particularly useful. Therefore, in these networks, the stub feature on spokes is usually combined with route filtering and summarization on the hub router. The hub router can be configured to advertise only the default route to the spoke router(s), filtering out all other more specific route entries, effectively reducing the routing table on the spoke to a single EIGRP-learned default route entry.

12.2.12. EIGRP Stub Routing Leak Map Support

12.2.13. Protocol-Dependent Modules

TODO

12.2.14. Goodbye Message and Graceful Shutdown

- broadcast when an EIGRP routing process is shut down
- Speeds convergence as peers don't have to wait the hold timer expiration
- Hello Message with all K-values set to 255
- Normal message displayed by routers that support Good Bye message

```
*Apr 26 13:48:42.523: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.1
(Ethernet0/0) is down: Interface Goodbye received
```

- Misleading message displayed by router which doesn't support the Goodbye message

```
*Apr 26 13:48:41.811: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor
(Ethernet0/0) is down: K-value mismatch
```

- The receipt of a goodbye message by a non supporting peer does not disrupt normal network operations.
- The nonsupporting peer will terminate the session when the hold timer expires
- The sending and receiving routers will converge normally after the sender reloads

12.2.15. Summarization

- All subnets are suppressed
- Creates boundary for Query propagation
    - If a router receives a Query for a network it does not have in its topology table, it will

immediately send back a Reply indicating an unreachable destination, without itself going active and propagating the Query further.

Task: Enable auto-summarization

```
(config-router)# auto-summarization
```

- Cannot be used in divergent networks
- create null0 summary

Task: Advertise a single summary in EIGRP classic mode

```
(config-if)# ip summary-address eigrp <asn> <prefix> <mask>
```

Task: Advertise a single summary in EIGRP named mode

```
(config-router-af-interface)# summary-address <prefix> <mask>
```

Task: Configure summarization to advertise a default route into EIGRP

```
(config-if)# ip summary-address eigrp <asn> 0.0.0.0 0.0.0.0
```

- All subnets will be suppressed because all IPv4 networks are subnet of 0/0

Task: Configure a fixed metric for EIGRP summary address

```
(config-router)# summary-metric <network-address> <subnet-mask>
                                 { <bandwidth> <delay> <reliability> <load> <mtu> [
distance <ad>  ] | distance  <ad>}
```

When EIGRP creates a summary route, it includes a metric with the route in order to advertise it. EIGRP searches for components of the summary to be suppressed and represented by the summary. EIGRP finds the component with the best metric and copies the metric from the component into the summary. Components of the summary may change often, which means that every time the best component metric changes, the summary needs to be readvertised to all its peers. Even if the best component metric is not the one that changed, EIGRP still has to search every topology entry to make sure the summary is not affected. This can add a significant processing overhead.

Leak map

Task: Advertise specific subnets of a EIGRP summary

```
(config-if)# ip summary-address eigrp <asn> <prefix> <mask> leak-map <route-maps>
```

Floating Summary Routes

TODO - By default, summarization install a route to Null0 to match the summary to prevent forwarding traffic for unreachable destinations. -

Poisoned Floating Summarization

TODO

12.2.16. EIGRP Route Authentication

- Supports MD5 in classic mode
- Supports MD5 and SHA-256 in multi-af mode

Task: Use MD5 password in EIGRP classic mode

```
(config-if)# ip authentication mode eigrp <asn> md5
(config-if)# ip authentication key-chain eigrp <asn> <password>
```

Task: Use MD5 password in EIGRP named mode

```
(config-router-af-interface)# authentication mode md5
(config-router-af-interface)# authentication key-chain <sesame>
```

Task: Authenticate EIGRP neighbor with SHA-256 password

```
(config-router-af-interface)# authenticate mode hmac-sha-256 <password>
```

- Can be applied at the **af-interface-default** in multi-af mode

12.2.17. Link Bandwidth Percentage

- by default, EIGRP packets consume max 50% of the link bandwidth as configured by the **bandwith** command
- bandwidth configured by **bandwidth** in AS configuration and **bandwith-percent** for named configuration

12.2.18. EIGRP autonomous system configuration

Task: Create a basic EIGRP AS system configuration

```
(config)# router eigrp asn
(config-router)# network a.b.c.d [e.f.g.h]
```

- A maximum of 30 EIGRP can be configured
- EIGRP sends updates only interfaces in the specified networks

Task: Verify eigrp topology

```
show ip eigrp topology [all-links]
show ip eigrp topology [prefix/len]
```

12.2.19. Router ID

- Used to avoid routing loops
- Advertised inside internal and external routes (in later IOS)
- same rule as OSPF

Task: Specify the EIGRP Router ID

```
(config-router)# eigrp router-id <a.b.c.d>
```

0.0.0.0 and 255.255.255.255 are not allowed

12.2.20. Unequal Load Balancing

If CD is the Computed Distance, then the eligible Feasible successor must satisfy the inequality below:

```
CD  via Successor < CD via Feasible Successor < variance * CD via Successor
```

The amount of traffic flowing over a particular path can be computed as this ratio:

```
Highest Installed Path Metric / Path Metric
```

- The unequal-cost paths installed into the routing table also count toward the maximum number of parallel paths to a destination configured using the maximum-paths command. Depending on your network topology and requirements, it might be necessary to modify this setting.

Task: Enable EIGRP unequal load balancing

```
(config-router)# variance <number>
```

Task: Enable EIGRP unequal load balancing in named configuration

```
(config-router-topology)# variance <number>
```

12.2.21. Add-Path Support

- Allow a Hub (dual-homed in DMVPN) to advertise multiple-equal cost routes to the same destination
  - must have the multiple equal-cost installed in its routing table
  - must disable Split Horizon on the tunnel towards the spokes
  - must have variance = 1, no unequal load balancing on the hub and the spokes
  - must desactived **next-self-hop [no-ecmp-mode]**
  - must be configured in the af-interface section of the named mode configuration
  - In certain scenarios, such as DMVPN deployments in which multiple branch offices are dual homed, hub routers usually have information about both routes to a particular dual-homed branch office, and can perform equal-cost load balanc ing on their end. However, without an additional mechanism, a hub is unable to advertise these equal-cost routes to other spoke routers. As a result, the other spokes only see a single route to the dual-homed branch office without an ability to perform load balancing over multiple paths, and if the single route they know about fails, they need to go over the usual reconvergence process in EIGRP to learn about the other route.
  - Spoke routers do not need to be specifically configured for the Add-Path feature, apart from possible tuning of the maximum-paths command to be allowed to insert multiple equal-cost paths into their routing tables.

12.2.22. Passive Interface

- Suppresses EIGRP hello packets and routing updates on interfaces
  - Doesn’t form adjancencies
  - Includes the interface addresses in the topology database

Task: Configure EIGRP passive interfaces

```
(config-router)# passive-interface [default] [<interface-type>  <interface-number>]
```

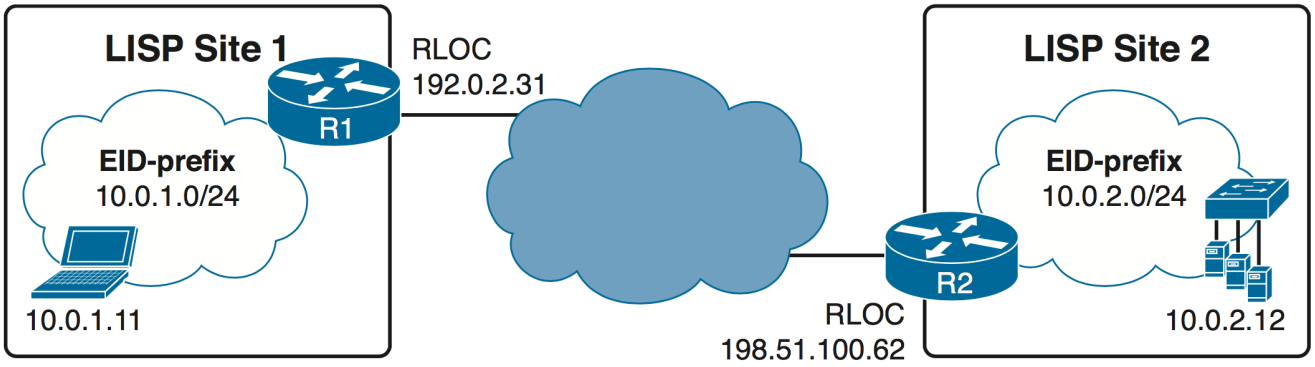
12.2.23. EIGRP Over The Top

- Enables a single end-to-end routing domain between two or more EIGRP sites that are connected using a private or a public WAN connection.
- Relies on LISP
- Benefits:
  - no dependency on the type of WAN connection used.

- no dependency on the service provider to transfer routes.
- no security threat because the underlying WAN has no knowledge of enterprise routes.
- simplifies dual carrier deployments and designs by eliminating the need to configure and manage EIGRP-BGP route distribution and route filtering between customer sites.
- allows easy transition between different service providers.
- supports both IPv4 and IPv6 environments.

LISP

- Locator/Identifier Seperation Protocol
- Separate the identity and location into two independent entities, each of them represented by a complete address, and provide a mapping service so that the address representing the identity of a host can be resolved into the address that represents its location.
- Uses EID (EndPoints Identifiers) and RLOC (Routing Locator) -



- LISP hence has both a control and a data plane. The control plane in LISP comprises the registration protocol and procedures by which the tunnel routers R1 and R2 register the EIDs they are responsible for along with their RLOCs in a LISP-mapping service, and using these registrations they map EIDs into RLOCs. The data plane defines the actual tunnel encapsulation used between Routers R1 and R2 when two hosts from each LISP sites communicate.
- In OTP, EIGRP serves as the replacement for LISP control plane protocols. Instead of doing dynamic EID-to-RLOC mappings in native LISP-mapping services, EIGRP routers running OTP over a service provider cloud create targeted sessions, use the IP addresses provided by the service provider as RLOCs, and exchange routes as EIDs.
- OTP is based on creating targeted EIGRP sessions between customer edge routers, and using the routing information carried by EIGRP to populate both routing tables and LISP mapping tables. The edge routers do not exchange any routing information with the ser- vice provider routers. Thus, this solution is fully controlled by a customer and requires no cooperation with the service provider, apart from providing full IP connectivity between customer routers

OTP CE

Task: Configure EIGRP OTP on CE

```
(config)# router eigrp test
(config-router)# address-family ipv4 unicast autonomous-system 100
(config-router-af)# neighbor 10.0.0.2 gigabitethernet 0/0/1 remote 3 lisp-encap 1
(config-router-af)# network 192.168.0.0
(config-router-af)# network 192.168.1.0
```

OTP Route Reflectors

Task: Configure EIGRP Route Reflectors

```
(config)# router eigrp test
(config-router)# address-family ipv4 unicast autonomous-system 100
(config-router-af)# af-interface gigabitethernet 0/0/1
(config-router-af-interface)# no next-hop-self
(config-router-af-interface)# no split-horizon
(config-router-af-interface)# exit
(config-router-af)# remote-neighbors source gigabitethernet 0/0/1 unicast-listen lisp-
encap 1
(config-router-af)# network 192.168.0.0
```

EIGRP Logging and Reporting

Task: Display the contents of the EIGRP log

```
# sh ei address-family {ipv4 | ipv6} events
```

Task: configure EIGRP logging

```
Router(config-router)# eigrp ?
event-log-size : Set max log size (default=500)
event-logging  : Log IP-EIGRP routing events (default)
log-neighbor-changes : enable IP-EIGRP neighbor logging (default)
log-neighbor-warnings :  Enable/Disable IP-EIGRP neighbor warnings (default=every
10seconds)
```

12.3. OSPF

Configuration Guides | IP Routing | OSPF

- link-state interior gateway protocol
- RFC 2328
- Dijkstra short path first algorithm
- classless protocol
- Transport via IP protocol 89

    - multicasts to 224.0.0.5 for AllSPF routers and to 224.0.0.6 for Designated Routers
    - unicasts
- equal-cost multipath
- hierarchical design to reduce traffic
- authentication updates

12.3.1. Neighbors

To form adjacency neighbors must agree on

- unique router ID
- unique interface IP address
    - primary IP address for OSPFv2
    - link-local address for OSPFv3
- common attributes
    - interface area-id
    - authentication
    - hello and dead timers
    - area type (normal, stub, NSSA,)
    - interface MTU
    - other optional capabilities

12.3.2. Router id

Determined by these rules in order of preference at boot or ospf process restart:

- manually configured router id
- highest IP address of an up/up loopback not used by other OSPF process
- highest IP address of an up/up non-loopack interfaces not used by other OSPF process

Task: Set the router-id

```
(config-router)# router-id <a.b.c.d>
```

12.3.3. DR Election

- There is no pre-emption in ospf
    - Router must wait for the failure of the current DR
    - Use the WAIT timer = DEAD timer

- on hub-and-spoke, best practice is to have hub as DR and spokes not eligible as DR with priority=0

Task: Priority

```
(config-if)# ip ospf priority <0-255>
```

Task: Set the WAIT timer

```
(config-if)# ip ospf dead-timer <seconds>
```

12.3.4. Ospf cost

"Cost" = 10^8 / "Bandwidth(bps)"

Task: Description

```
(config-router)# auto-cost reference-bandwitch <bps>
```

12.3.5. OSPF Packet Format

Common OSPF Packet Header

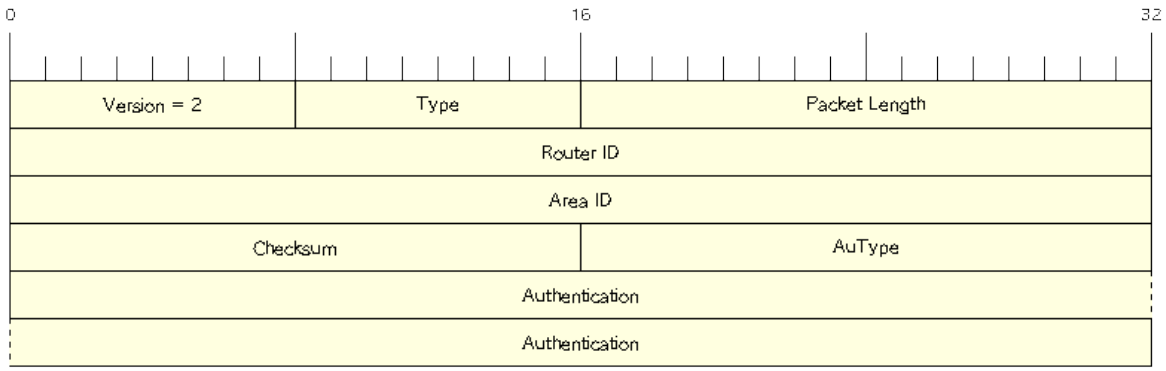


Figure 19. OSPF header format

Type

Hello (1), database description (2), Link-State Request (3), Link-State Update (4), or Link-State Acknowledgment (5).

Packet length

Length of the protocol packet in bytes including the OSPF header.

Router ID

The ID of the router originating the packet.

Area ID

The area that the packet is being sent into.

Checksum

standard IP checksum of the entire contents of the packet, excluding the 64-bit authentication field.

AuType

Identifies the authentication scheme to be used for the packet.

- 0: no authentication
- 1: plain-text authentication
- 2: cryptographic authentication

Authentication

64-bit field for use by the authentication scheme.

Hello Packet

- Sent from the primary IP address (not the secondary addresses)
- Every 10 seconds (Ethernet), 30 seconds (Non-broadcast)

OSPF neighbors will become fully adjacent if one or both of the neighbors are using unnumbered interfaces for the connection between them.

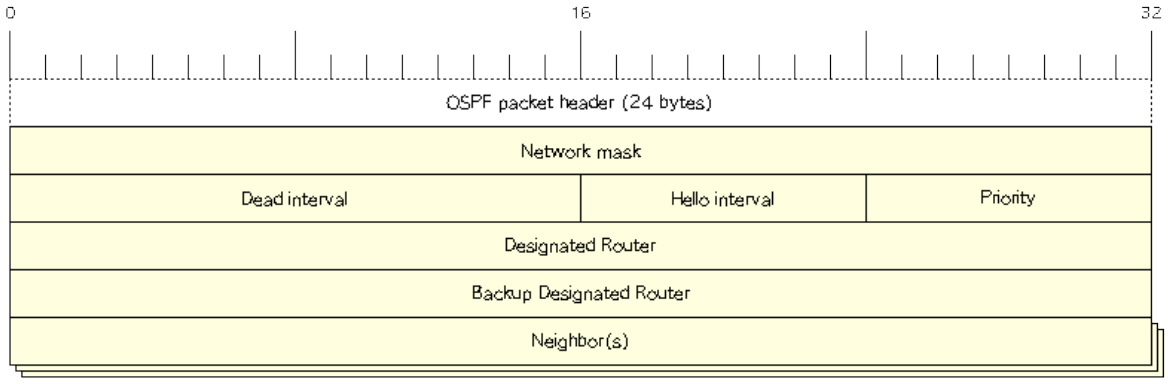


Figure 20. OSPF Hello Packet format

Task: Configure OSPF Hello interval

```
(config-if)# ip ospf hello-interval <seconds>
```

Task: Set the interval during which at least one OSPF hello packet must be received from a neighbor before the router declares that neighbor down

```
(config-if)# ip ospf dead-interval {<seconds> | minimal hello-multiplier <number>}
```

Database Description Packet

- Uses an OSPF-defined simple error-recovery process.
  - Each DD packet, which can contain several LSA headers, has a sequence number assigned.
  - The receiver acknowledges a received DD packet by sending a DD packet with the identical sequence number back to the sender.
  - The sender uses a window size of one packet and then waits for the acknowledgment before sending the next DD packet.
- Only the master is allowed to send DD packets on its own accord as well as to set and increase their sequence numbers.
- A slave is allowed to send a DD packet only as a response to a DD packet received from master router, and must use the same sequence number. In effect, a slave is polled by the master and only responds to it.
  - If a slave has more DD than the master, he uses the M flag

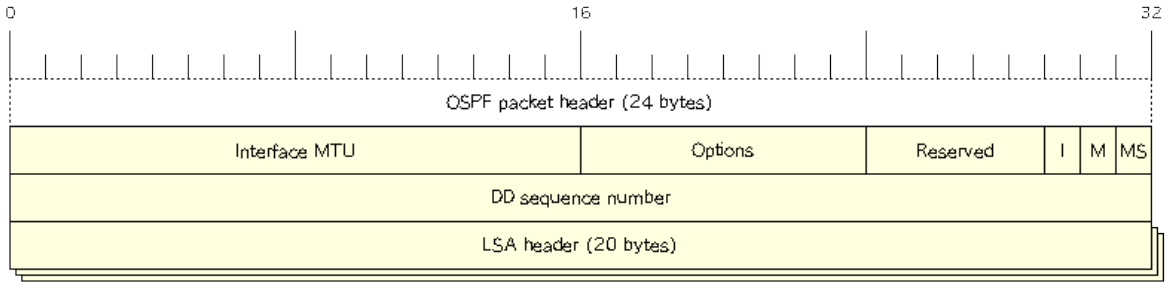


Figure 21. OSPF DD Packet format

Interface MTU

Size of the largest IP message that can be sent on this router's interface without fragmentation

Options

For optional OSPF capabilities

I-bit

Initial for the first in a sequence of DD messages

M-bit

More DD follow this one

MS-bit

if this message is sent by the master in the communication

Link State Request

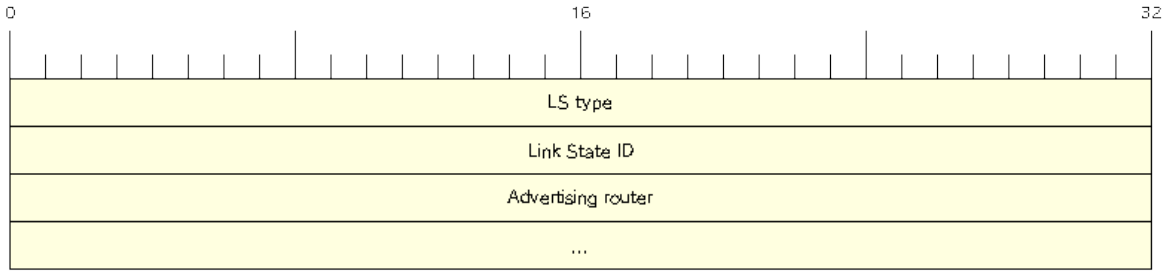


Figure 22. OSPF Link State Request format

Link State Update

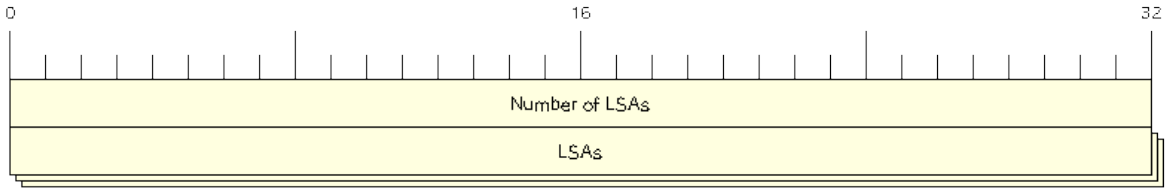


Figure 23. OSPF Link State Update format

Link State Acknowledgment

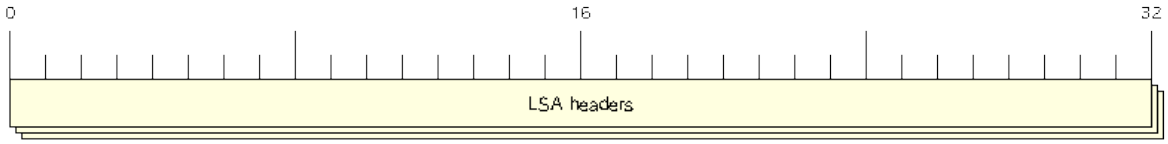


Figure 24. OSPF Link State Acknowledgment format

LSA headers

Contains LSA headers to identify the LSAs acknowledged.

Link-State Packets

- only a router that has originated a particular LSA is allowed to modify it or withdraw it.
  - Other routers must process and flood this LSA within its defined flooding scope if they recognize the LSA's type and contents, but they must not ever change its contents, block it, or drop it before its maximum lifetime has expired.
- has a unique LSID (Link State Identifier)

Type 1

Router LSA

- one per router per area
- lists the RouterID, the IP Addresses and neighbors for each interface in that area

- represents Stub networks (subnet on which a router has not formed any neighbor relationships)
- flooded only within the same area
- LSID = Router ID

Type 2

Network LSA

- one per transit network
  - network over which two or more OSPF routers have become neighors and elected a DR so that traffic can transit between them
  - except for point-to-point connection treated as a combination of p2p link and a stub IP network (to facilate unnumbered p2p links)
- generated by DR
- describes the set of routers attached to a particular network
- describes the subnet and the router interfaces connected to the subnet
- flooded only within the area that contains the network
- LSID = DR's interface IP Address on that subnet

Type 3

Summary inter-area LSA

- Generated by ABR
- describes inter-area routes to network
  - represents networks present in one area when being advertised into another area.
  - Defines the subnets in the origin area, and cost, but no topology data.
- Flooded only within its area of origin; reoriginated on ABRs.

Type 4

Summary inter-area LSA

- Generated by ABR
- Flooded by ABR to all areas except the area containing the ASBR
- describes routes to ASBR
  - tells other routers in the area how to get to the advertising router of an external route
- Flooded all areas except the area containing the ASBR

Type 5

AS external LSA

- originated by ASBR

- describes routes to destinations external to the AS
- flooded all over except stub areas

Type 6

Group Membership LSA

- defined for MOSPF
- Not supported by Cisco

Type 7

NSSA External LSA

- Created by ASBRs inside an NSSA, instead of a type 5 LSA.
- Flooded only within its area of origin;
- converted to type 5 LSA on an ABR toward other areas.

Type 8

External Attributes LSA

- Created by ASBRs during BGP-to-OSPF redistribution to preserve BGP attributes of redistributed networks.
- Not implemented in Cisco routers

Type 9-11

Opaque LSA

- Used as generic LSAs to allow for easy future extension of OSPF;
  - for example, type 10 has been adapted for MPLS traffic engineering.
- have different flooding scope:
  - Type 9 has link-local flooding scope,
  - type 10 has area-local flooding scope,
  - type 11 has autonomous system flooding scope equivalent to the flooding scope of type 5 LSAs (not flooded into stubby areas and NSSAs).
    1. OSPF's SPF algorithm links different pieces of information together.

For a router in Area 1 to reach the external route in Area 3, it has to look at the Type-5 that represents the external route. Then it has to look at the Type-4 representing the ABR on the area that the ASBR lives in. Then we have to look at the Type-3 to get to that remote ABR. Finally we look at the Type-1 and Type-2 LSAs in our area to determine how to get to our closest ABR.

Read more here.

Task: Display the OSPF database

```
# sh ip ospf database
```

12.3.6. Backbone

ABR

Router actively connected to multiple areas **including** Area 0

- has one LSDB for each area
- runs the SPF for each LSDB then combines the result in a single routing table
- can summarize and filter routes
- ignores type 3 LSAs learned in a nonbackbone area during SPF calculation, which prevents an ABR from choosing a route that goes into a nonbackbone area and then back into the backbone.

12.3.7. Stubby Areas

All stubby area types - block Type 4/5 LSA - automatically inject default routes except NSSA

Stubby area

- Doesn't have an ASBR

Task: Configure a stubby area

```
(config-router)# area <id> stub
```

Totally Stubby

- Stubby areas where Type 3 are blocked

Task: Configure totally stubby areas on the ABR

```
(config-router)# area <id> stub no-summary
```

NSSA

- Contains one or more ASBRs
- Allows creation of Type 7
- Doesn't automatically inject default routes
- The ABR with highest RID translates Type 7 to Type 5

Task: Configure NSSA

```
(config-router)# area <id> nssa
```

Task: Inject default routes in NSSA

```
(config)# area <id> nssa default-information-originate
```

Totally NSSA

- NSSA where Type 3 are blocked

Task: Configure Totally NSSA

```
(config-router)# area <id> nssa no-summary
```

12.3.8. OSPF path selection

- Intra-Area > Inter-Area > External Routes (E1/N1 > E2/N2)

12.3.9. Virtual links

- purposes:
  - Areas not physically connected to area 0
  - partitioning the backbone
- transit area can not be stub

Router A

```
(config)# router ospf 10
(config-router)# area 2 virtual-link 2.2.2.2
```

Router B

```
(config)# router ospf 10
(config-router)# area 2 virtual-link 1.1.1.1
```

Task: TODO

```
(config-router)# no capability transit
```

Task: Configure Authentication on virtual links

```
! Null
(config-router)# area <id> virtual-link <router-id> authentication { null }

! Plaintext
(config-router)# area <id> virtual-link <router-id> authentication { authentication-
key <key-value> }

! MD5
(config-router)# area <id> virtual-link <router-id> authentication { message-digest
message-digest- key key-num md5 key-value}

! Cryptographic
(config-router)# area <id> virtual-link <router-id> key-chain <key-chain-name>
```

What are ospf areas and virtual links

12.3.10. Network types

broadcast

- multicast hellos every 10 seconds
- automatic neighbor discovery
- DR/BDR election
- default for LAN ethernet, TR, FDDI
- DR doesn't change the next hop of advertised prefixes

Point-to-point

- only 2 routers
- automatic neighbor relationships
- no DR/BDR election
- multicast hellos every 10 seconds
- default for HDLC and PPP

Non-broadcast

- unicast hellos every 30 seconds
- manual configuration of neighbor
- DR/BDR election
- default on Frame Relay, X.25 and SMDS

Point-to-multipoint

- multi-access, broadcast
- hellos every 30 seconds

- automatic discovery of neighbor (MA)
- DR/BDR election
- one IP subnet
- maintain connectivity during a VC failure ???
- generates host routes (with mask /32) for each neighbor
- default for ???

Point-to-multipoint non-broadcast

- manual configuration of neighbor
- no DR/BDR election
- network proprietary to Cisco
- hellos every 30 seconds

Loopback

if Multi-Access network type then DR/BDR election

if non-broadcast then manual configuration of neighbors

OSPF design guide: selecting interface network types

Task: Configure OSPF network type

```
(config-if)# ospf network {broadcast| point-to-point| point-to-multipoint [non-
broadcast] | non-broadcast | loopback }
```

12.3.11. Graceful restart

- enables a router to continue to forward packets during a restart of the routing process
- must be configured on all neighbor routers
- can also work with EIGRP, BGP, IS-IS
- default since IOS 12.4(6)T
- 2 versions: RFC 3623 and Cisco NSF

Cisco NSF

12.3.12. SPF throttling

12.3.13. capability vrf-lite

Read OSG, chapter 19, VRF lite, pp. 872-876

http://www.cisco.com/en/US/docs/ios-xml/ios/iproute_ospf/command/ospf-a1.html#wp2582896905

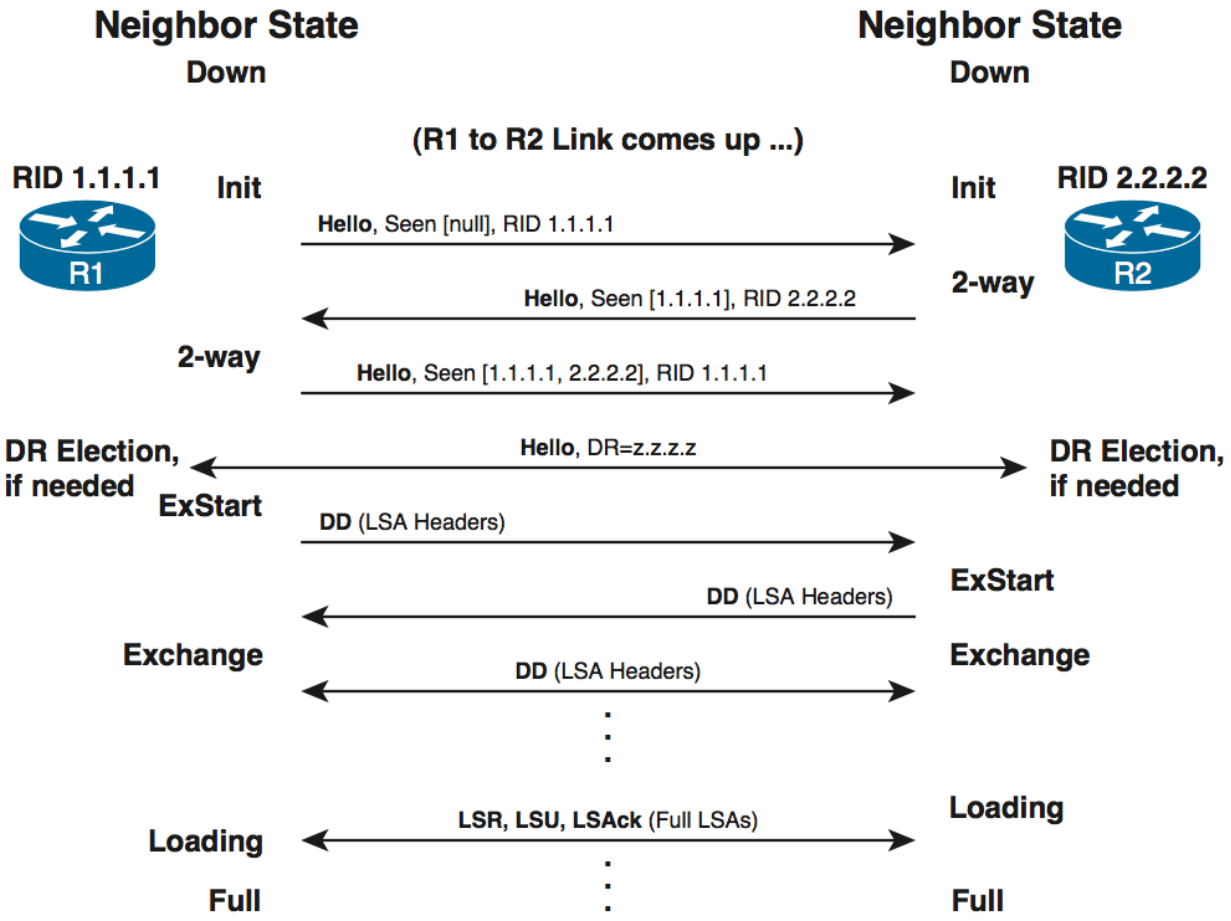
12.3.14. summarization

Why the null 0 interface is added ?

- do prevent routing loops
  - packets destined for the routes that have been summarized will a longer match
  - packets destined to summary routes will be dropped

See good explanation

12.3.15. OSPF states



Down

- No hellos have been received from neighbors

Attempt

- Unicast hello packet has been sent to neighbor, but not yet received back
- only used for manually configured NBMA neighbors

Init

- I have received a hello packet from a neighbor, but they have not acknowledged a hello from me

2-way

- I have received a hello packet from a neighbor and he acknowledged a hello from me
- I can see my Router Id in the neighbor's hello packet
- Stop here for DROthers

Exstart

- Master & slave relationship is formed where master has higher Router-id
- Master chooses the starting sequence number of the DBD packets that are used for actual LSA exchange.

Exchange

- Local link state database is sent through DBD packets
- DBD sequence number is used for reliable acknowledgement/retransmission

Loading

- LSR packets are sent to ask for more info about a particular LSA

Full

- Neighbors are fully adjacent and databases are synchronized.

Key Point

In the beginning of the exchange, each router places the other into the ExStart state. Each of them considers itself to be the master, and sends an empty DD packet to the other router, containing a randomly chosen sequence number, and MS (Master), M (More), and I (Init) flags set to 1. After receiving the neighbor's DD packet, however, the router with the lower RID will change its role to slave, and it will respond with a DD packet with MS and I flags cleared and the sequence number set to the sequence number of master's DD packet. This accomplishes the master/slave selection, and both routers move to the Exchange state. The master will then send a DD packet with the sequence number incre- mented by 1, optionally containing one or more LSA headers, and the slave will respond with a DD packet reusing the same sequence number from the received packet, optionally advertising its own LSA headers. The exchange continues in the same fashion, with the master incrementing the sequence number of each subsequent DD packet, until both routers have advertised all known all LSA headers (the master will stop sending DD pack- ets when it has advertised all LSA headers itself and the last DD response from the slave has the M flag cleared).

ospf design guide: link-state advertisements

12.3.16. OSPF process

Task: Enable OSPF process (legacy command)

```
(config)# router ospf <process-id>
(config-router)# network <a.b.c.d> [<w.i.l.d>] area <id>
```

- inject both the primary and secondary addresses
- If an interface is IP unnumbered, and there is a **network** statement that matches the IP address of the primary interface, inject both the primary interface and the unnumbered interface

Task: Enable OSPF Process (interface level)

```
(config-if)# ip ospf <process-id> area <id>
```

- inject any and all secondary subnets

Task: Prevent OSPF to advertize secondary prefixes

```
(config-if)# ip ospf <process-id> area <id> secondaries none
```

12.3.17. OSPF authentication

Classic OSPF Authentication

- Null , default: type 0
- Plain-text, simple password authentication

```
(config-router)# area <id> authentication
(config-if)# ip ospf authentication-key <string>
```

- Message digest authentication

```
(config-router)# area <id> authentication message-digest
(config-if)# ip ospf message-digest-key <key-id> md5 <string>
```

Key rollover procedure with Multiple MD5 keys

Multiple MD5 keys with different key IDs are allowed per interface. This allows for graceful key migration where a new key can be added without disrupting the adjacencies.

- To sign sent packets, it always uses the key that was added as the last one to the interface (regardless of the key number).
- To authenticate the received packet, it uses the key ID that is indicated in the packet.
- If a neighbor is detected on an interface that uses a different key number than this router, OSPF enters a key migration phase in which it sends all packets as many times as how many keys are configured on the interface, and each packet is signed with a different key.
- The migration phase ends when all neighbors have migrated to the same key as the one used to sign sent packets by this router.
- This procedure is also called the OSPF key rollover procedure.
- Because plaintext passwords do not have key numbers, the key rollover is not available for plaintext authentication.

Extended Cryptographic OSPF Authentication

- Uses SHA-HMAC (Secure Hash Algorithm - Hash Message Authentication Code) as per RFC 5709
- Uses key chains
  - Each key in the key chain must have a cryptographic algorithm configured using a per-key **cryptographic-algorithm** command. Failure to do so will result in OSPF not using that key.
  - Each key in a key chain can be configured with the **send-life-time** and accept-life-time keywords to limit its usability to a particular timeframe. If multiple keys in the key chain are eligible to sign egress packets, the key with the highest key ID will be used. Be aware that this behavior differs from RIPv2 and EIGRP that select the key with the lowest key ID.
  - The key rollover procedure as used by classic OSPF is not used with key chains. There is no key migration phase of sending multiple OSPF packets signed with different valid keys.
    - To sign egress packets, use the valid key with the highest key ID in the key chain.
    - To authenticate ingress packets, try to use the key indicated in the received packet.

Task: Configure a cryptographic algorithm for the key chain

```
(config)# key-chain <name>
(config-keychain)# key <number>
(config-keychain-key)# cryptographic-algorithm ?

  hmac-sha-1    HMAC-SHA-1 authentication algorithm
  hmac-sha-256  HMAC-SHA-256 authentication algorithm
  hmac-sha-384  HMAC-SHA-384 authentication algorithm
  hmac-sha-512  HMAC-SHA-512 authentication algorithm
  md5           MD5 authentication algorithm
```

Task: Configure the extended cryptographic OSPF authentication

```
(config-if)# ip ospf authentication key-chain <key-chain-name>
```

Configuring the extended cryptographic authentication using the area OSPF process level command is not supported.

12.3.18. TTL Security Check

- Drops packets with TTL < 255 except on virtual links and sham links
  - If all OSPF routers sent their packets with TTL set to 255, receiving an OSPF packet with its TTL less than 255 would be a clear indication that the packet originated outside the network segment over which it was received. Because OSPF com munication is, with the notable exception of virtual links and sham links, always based on direct router-to-router communication, receiving an OSPF packet outside a virtual link or a sham link with its TTL less than 255 is a possible indication of a malicious activity.

Task: Configure the Time-to-Live (TTL) security check feature on a specific interface

```
(config-if)# ip ospf ttl-security [hops <count> |disable]
```

Task: Configure the Time-to-Live (TTL) security check feature on all interfaces

```
(config-router)# ip ospf ttl-security all-interfaces
```

Task: Configure TTL security on a virtual link

```
(config-router)# area virtual-link ttl-security <hops>
```

Task: Configure TTL security on a sham link

```
(config-router)# area virtual-link ttl-security <hops>
```

12.3.19. SPF

spf timers

- spf-delay: between topology change notifications and recalculation of the shortest path
- spf-holdtime : between spf calculations

Task: Configure spf timers

```
(config-router)# timers spf seconds <seconds>
```

SPF Throttling

- Defines a variable-length wait interval between two consecutive SPF runs
- Controls by 3 parameters:
  - spf-start: initial wait interval before an SPF computation, if the network has been stable for a prolonged period of time.
  - spf-hold: wait time between subsequent SPF runs, and its value doubles for each consecutive SPF run.
  - spf-max-wait: maximum time between two SPF runs (that is, doubling the spf-hold value is capped at spf-max-wait), and also defines a period during which the network must be stable for the wait interval to be set back to spf-start and the spf-hold to its preconfigured value. If the network has been stable for the last spf-hold period but not for the entire spf-max-wait since the last SPF run, the wait interval returns to the spf-start value but the subsequent wait will still be set to twice the previous spfhold value.

Task: configure spf throttling

```
(config-router)# timers throttle spf <spf-start> <spf-hold> <spf-max-wait>
```

Task: Verify SPF throttling configuration

```
# sh ip ospf | i SPF

 Initial SPF schedule delay 10000 msecs
 Minimum hold time between two consecutive SPFs 15000 msecs
 Maximum wait time between two consecutive SPFs 100000 msecs
```

LSA Throttling

Task: Configure LSA Throttling

```
(config-router)# timers throttle lsa all <start-interval> <hold-interval> <max-
interval>
```

Task: Verify LSA Throttling configuration

```
# sh ip ospf | i LSA

 Initial LSA throttle delay 10000 msecs
 Minimum hold time for LSA throttle 15000 msecs
 Maximum wait time for LSA throttle 100000 msecs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
```

TODO Apart from throttling the LSA origination, a router can also be configured to ignore the same LSA upon arrival if it appears to arrive too often. This throttling of arriving LSAs is configured using the timers lsa arrival milliseconds OSPF command. If two or more same LSAs arrive less than milliseconds apart, only the first one is accepted and the remaining LSAs are dropped. In effect, the same LSA is accepted only if it arrives more than milliseconds after the previous accepted one. The default setting is 1000 millisec- onds and can be seen in the show ip ospf output in Example 9-16. Obviously, the value of the minimum LSA arrival interval should be smaller than the neighbors' initial hold interval in LSA Throttling. Otherwise, a neighbor would be allowed to send an updated LSA sooner than this router would be willing to accept it.

Incremental SPF

Task: Configure Incremental SPF

```
(config-router)# ispf
```

Task: Verify Incremental SPF configuration

```
# sh ip ospf | i Incremental

 Incremental-SPF enabled
```

12.3.20. OSPF Filtering

Routes Filtering not LSA filtering

- uses **distribute-list**
- The distribute list in the inbound direction applies to results of SPF—the routes to be installed into the router's routing table.
- The distribute list in the outbound direction applies only to redistributed routes and only on an ASBR; it selects which redistributed routes shall be advertised.
- The inbound logic does not filter inbound LSAs; it instead filters the routes that SPF chooses to add to that one router's routing table.
- If the distribute list includes the incoming interface parameter, the incoming interface is checked as if it were the outgoing interface of the route.

ABR Type 3 LSA filtering

- allows an ABR to filter type 3 LSAs at the point where the LSAs would normally be created.

Task: Filter Type 3 LSA on the ABR

```
(config-router)# area <id> filter-list prefix <prefix-list-name> { in | out }
```

Using the area range no-advertise option

Task: Summarize and do not advertise components

```
(config-router)# area <id> range <prefix /length> not-advertise [ cost cost ]
```

12.3.21. OSPFv2 Prefix Suppression

- RFC 6860 defines a method of hiding, or suppressing, the transit link prefixes in OSPF TODO Complete this

Task: Activate OSPFv2 Prefix Suppression for the entire router

```
(config-router)# prefix-suppression
```

suppress all prefixes on all its OSPF-enabled interfaces except loopbacks, secondary IP addresses, and prefixes on passive interfaces. Such prefixes are considered nontransit prefixes.

Task: Activate OSPFv2 Prefix Suppression on a specific interface

```
(config-if)# ip ospf prefix-suppression [disable]
```

12.3.22. OSPF Stub Router

- allows a router to either temporarily or permanently be prevented from becoming a transit router.
  - a transit router is simply one to which packets are forwarded, with the expectation that the transit router will forward the packet to yet another router.
  - a nontransit routers only forward packets to and from locally attached subnets.

TODO Better explanation

12.3.23. OSPF Graceful Restart

12.3.24. OSPF Graceful Shutdown

12.4. IS-IS

12.5. BGP

Configuration guides | IP Routing | BGP

- Exterior gateway protocol
- Creates loop-free inter-domain routing between AS.
- Path vector algorithm = (distance vector + AS-path loop detection)
- TCP 179
- AD: external 20 , internal and local 200
- RFC 1771

12.5.1. BGP message format

- Minimum size: 19 bytes
- Maximum size: 4096 bytes (why?)

BGP header

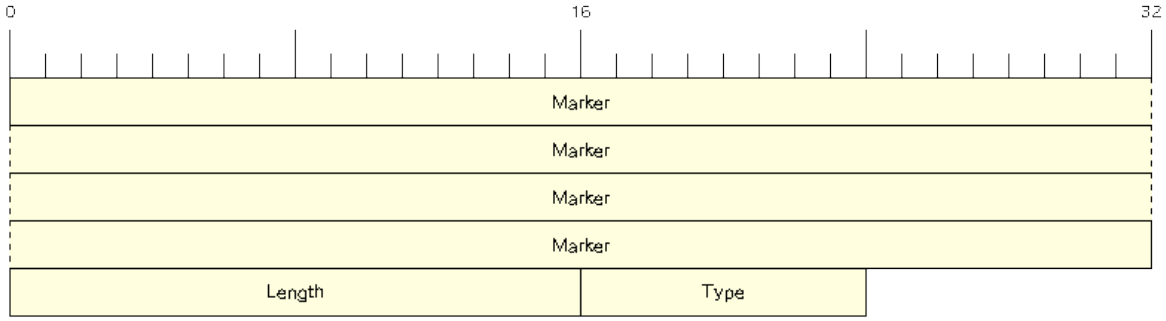


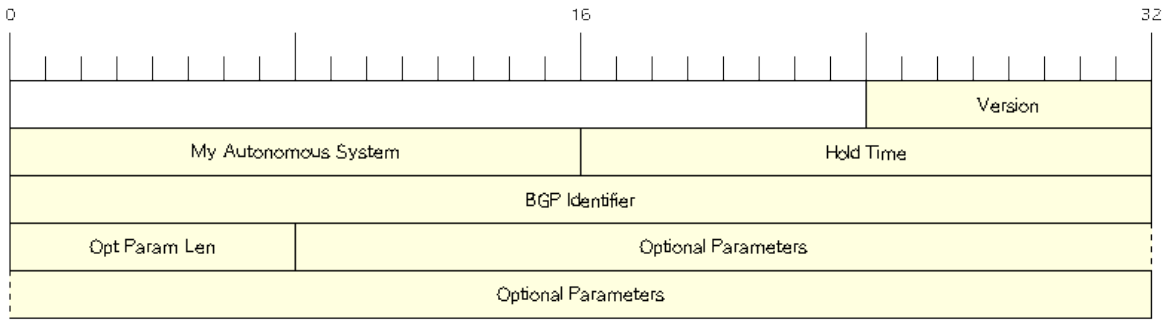
Figure 25. BGP header format

- Marker:
  - 16 bytes
  - set to all 1s for OPEN message or if OPEN message without authentication
  - computed by the authentication process
- Length:
  - 2 bytes
  - total length in bytes of the message including the header
- Type:
  - 1 byte

- indicates message type (1: Open, 2: Update, 3: Notification, 4: Keepalive)

OPEN

- Initiates the session
- Contains BGP version , local AS number, BGP router Id



- Version: 1 octet
- My autonomous system:
- Hold time:
  - maximum interval in seconds between successive Keepalive or Update messages.
  - A receiver compares the value of the Hold Time and the value of its configured hold time and accepts the smaller value or rejects the connection.
  - Can be set to zero to indicates that the connection is always up //find a better formulation
  - if not set to zero, the minimum recommended hold time is 3 seconds
- BGP identifier:
  - router ID
  - determined by these rules in order of preference at boot or bgp process restart:
    - manually configured router id
    - highest IP address of an up/up loopback
    - highest IP address of an up/up non-loopack
- Optional parameters length:
- total length in octects of the following Optional Parameters field
- Optional Parameters:
- Variable length field containing a triplet <Type: 1 octet,Length: 1 octet,Value>

Task: Configure the bgp router id

```
(config-router)# bgp router-id <ip-address>
```

KEEPALIVE

- Every 60 seconds
- Hold-time: 180 seconds

Task: Set the bgp network timers

```
(config-router)# timers bgp <keepalive-seconds> <holdtime-seconds>
```

Task: Set the bgp network timers for a specific neighbor

```
(config-router)# neighbor {<ip-address> | <peer-group-name>} timers <keepalive-
seconds> <holdtime-seconds>
```

UPDATE

- Advertises a single feasible route to a peer and/or withdraws multiple unfeasible routes

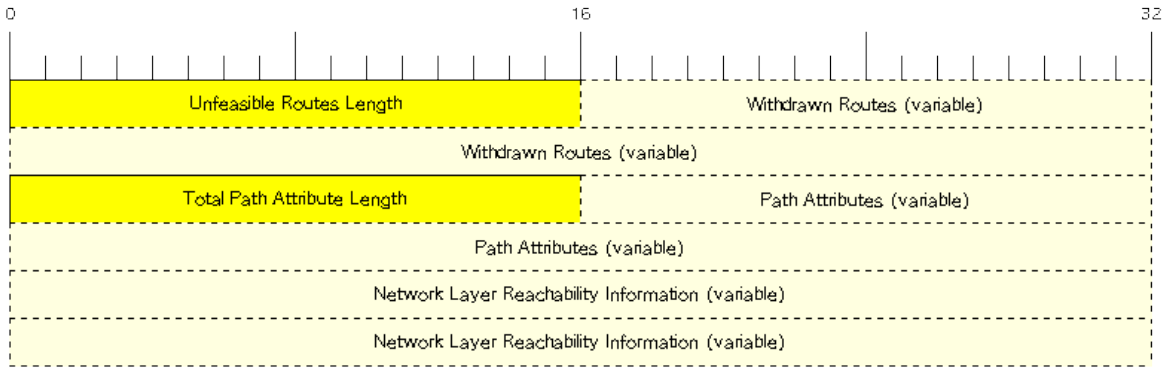


Figure 26. header format

- Unfeasible Routes Length
  - 2-octet field
  - total length of the following Withdrawn Routes field, in octets.
- Withdrawn Routes
  - variable-length
  - lists routes to be withdrawn from service.
  - Each route in the list is described with a (Length, Prefix) tuple in which the Length is the length of the prefix and the Prefix is the IP address prefix of the withdrawn route.
- Total Path Attribute Length
  - 2-octet
  - total length of the following Path Attribute field, in octets.
- Path Attributes

- variable-length
- lists the attributes associated with the NLRI in the following field. Each path attribute is a variable-length triple of (Attribute Type, Attribute Length, Attribute Value). The Attribute Type part of the triple is a 2-octet field consisting of four flag bits, four unused bits, and an Attribute Type code (see Attribute Type Code).

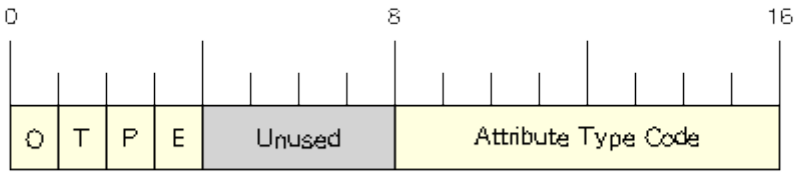


Figure 27. Attribute Type part of the Path Attributes field

Flag bits (1/0)

- O: Optional / Well-known
- T: Transitive / Non-transitive
- P: Partial / Complete
- E: Extended length / Regular length (2-bytes/ 1-bytes)
- U: Unused

Table 11. Attribute Type Code

Code	Attribute	Category
1	ORIGIN	Well-known mandatory
2	AS_PATH	Well-known mandatory
3	NEXT_HOP	Well-known mandatory
4	MULTI_EXIT_DISC	Optional nontransitive
5	LOCAL_REF	Optional transitive
6	ATOMIC_AGGREGATE	Well-known discretionary
7	AGGREGATOR	Optional transitive
8	COMMUNITY	Optional transitive
9	ORIGINATOR_ID	Optional nontransitive
10	CLUSTER_LIST	Optional nontransitive

tasks for Internet, no-export, no-advertise, local-as

NOTIFICATION

- go out in response to error, fatal condition

- torn down or reset the BGP peer session

BGP FSM States

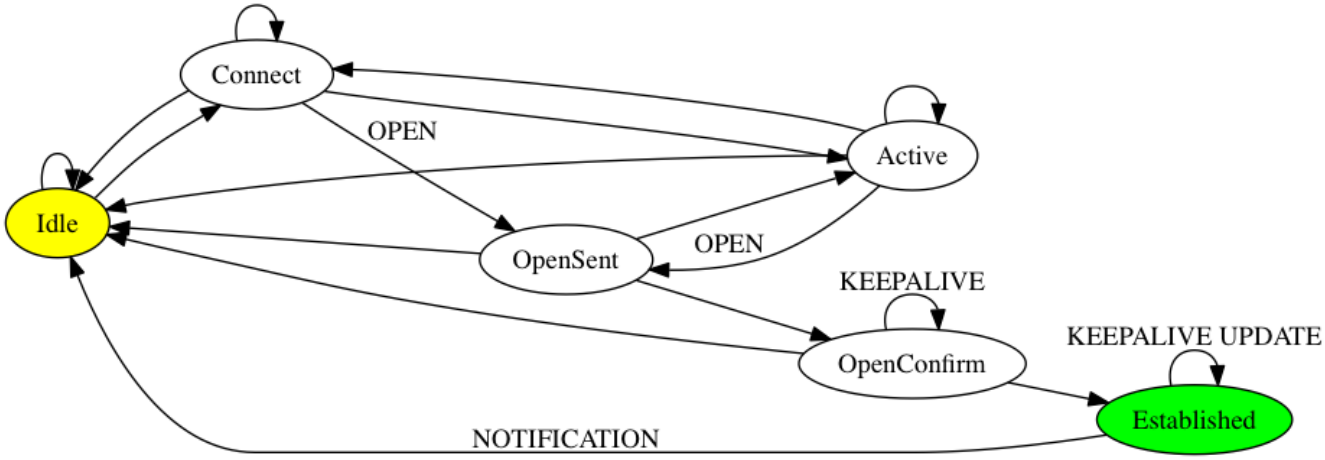


Figure 28. BGP neighbor negotiation finite state machines

- **Idle** – initial BGP state after enabling BGP process or resetting device.
- **Connect** - waits for a TCP connection with the remote peer. If successful, sends OPEN message. If not, resets the ConnectRetry timer and transitions to Active state.
- **Active** – attempts to initiate a TCP connection with the remote peer. If successful, sends OPEN message. If not, resets ConnectRetry timer and transitions back to Connect state
- **OpenSent** – TCP connection up and OPEN message sent, transition to OpenReceive state and wait for initial keepalive to move into OpenConfirm state. If TCP session disconnect, terminate BGP session, reset ConnectRetry timer, move back to Active State.
- **OpenConfirm** – OPEN messages sent and received. Wait for KEEPALIVE
- **Established** – KEEPALIVE received, neighbor parameters match. the BGP peer session is fully established. UPDATE messages containing routing information will now be sent.
- If peer stuck in **Active** state, potential problems can include:
  - no IP connectivity
  - incorrect **neighbor** statement
  - access-list filtering TCP port 179

TODO: To display transitions from idle to established with debug ip bgp

```
R1(config)# router bgp 123
R1(config-router)# no neigh 172.16.16.6 shutdown
*Mar 4 21:02:16.958: BGP: 172.16.16.6 went from
*Mar 4 21:02:16.958: BGP: 172.16.16.6 , delay 15571ms
*Mar 4 21:02:29.378: BGP: 172.16.16.6
*Mar 4 21:02:29.382: BGP: 172.16.16.6 rcv message type 1, length (excl. header) 26
*Mar 4 21:02:29.382: BGP: 172.16.16.6 rcv OPEN, version 4, holdtime 180 seconds *Mar 4
21:02:29.382: BGP: 172.16.16.6 went from
*Mar 4 21:02:29.382: BGP: 172.16.16.6   , version 4,   ,
holdtime 180 seconds
*Mar 4 21:02:29.382: BGP: 172.16.16.6     w/ OPTION parameter len: 16 BGP: 172.16.16.6
*Mar 4 21:02:29.382: BGP: 172.16.16.6 went from OpenSent to OpenConfirm
*Mar 4 21:02:29.382: BGP: 172.16.16.6 send message type 1, length (incl. header) 45
*Mar  4 21:02:29.394: BGP: 172.16.16.6 went from
```

12.5.2. Autonomous systems

- AS: set of routers under a single technical administration
- AS can be:
  - stub : only one exit
  - multihomed: multiple connections with the one or multiple providers
    - transit: allows traffic with origin and destination outside the AS
    - non-transit:

ASN format

- 2-byte (RFC 4271)
  - 0 - 65535
  - reserved: 0, 65535
  - public use: 1 - 64495
  - documentation: 64496-64511 (RFC 5398)
  - private use: 64512 - 65534
- 4-byte (RFC 5396)
  - Asplain: decimal value notation for 2-byte and 4-byte ASNs
  - Asdot: decimal value notation for 2-byte and dot notation for 4-byte ASN
  - Documentation: 65536-65551 (RFC 5398)
- AS 23456: reserved for gradual transition from 2-byte to 4-byte (RFC 4893)

Task: Modify the default output and regex match format for 4-byte ASN

```
(config-router)# bgp asnotation dot
```

12.5.3. BGP peers

- Manually configured and not automatically discovered
- Formed over a TCP connection
- Exchanges PA(Path Attributes) and NLRI (IP/prefix) with the same PA
- Starts with full BGP routing table then incremental updates
- Keeps table version number

iBPG peers

- same AS
- must be fully meshed within AS

eBGP peers

- different AS
- by default, one hop away but you can change that with **ebgp-multihop**

Task: Configure Neighbor

```
(config-router)# neighbor <ip-address> remote-as <asn>
```

Task: Enable the neighbor to exchange prefixes for the ipv4 unicast address family with the local device

```
(config-router)# address-family ipv4 [unicast | multicast | vrf <name>]
!TODO check the mode
(config-router)# neighbor <ip-address> activate
```

Task: Display info about the TCP and BGP connection to neighbors

```
# sh ip bgp neigbors <ip-address>
```

12.5.4. BGP peer groups

- Group of peers with the same update policies (outbound route maps, distribute lists, filter lists, update source ,)
- Benefits:
  - simplify configuration
  - make configuration updates more efficient

- Restrictions for eBGP peers:

Task: Create a BGP peer group

```
(config-router)# neighbor <peer-group-name> peer-group
```

Task: Assign a neighbor to a peer group

```
(config-router)# neighbor <ip-address> peer-group <name>
```

Task: Add a text description with a specified peer group

```
(config-router)# neighbor <peer-group-name> description <text>
```

Task: Disable a BGP peer or peer group

```
(config-router)# neighbor <ip-address> shutdown
```

12.5.5. BGP session reset

- Whenever the routing policy changes due to a configuration change
- Can be hard reset, soft reset or dynamic inbound soft reset

Task: Clear and reset BGP neighbor sessions

```
# clear ip bgp *
```

Task: Enable logging of BGP neighbor resets

```
(config-router)# bgp log-neighbor-changes
```

Task: Clear BGP update group membership and recalculate BGP update groups

```
# clear ip bgp update-group [ <index-group> | <ip-address> ]
```

Hard reset

- Tears down the peering sessions including the TCP connections
- Deletes prefixes learned from the peers.
- Pros: no memory overhead

Soft reset

- Stores prefix information

Do not tearn down existing peering sessions

- Can be configured for inbound or outbound sessions

Task:Configure a BGP speaker to perform inbound soft reconfiguration for peers that do not support the route refresh capability.

```
(config-router)# bgp soft-reconfig-backup
```

Task: Start storing updates for each neighbor that do not support route refresh

```
(config-router)# neighbor <ip-address|peer-group-name> soft-reconfiguration [inbound]
```

- All the updates received from this neighbor will be stored unmodified, regardless of the inbound policy. When inbound soft reconfiguration is done later, the stored information will be used to generate a new set of inbound updates.
- Memory requirements can increased.

Dynamic inbound soft reset

- Do not store update information locally
- Relies on dynamic exchanges with supporting peers
- The peers supports the capability if **show ip bgp neighbors** displays *Received route refresh capability from peer* .
- Use **bgp soft-reconfig-backup** to store updates for peers who do not support the refresh route capability

Routing policy change management

TODO: add this part under bgp reset

12.5.6. BGP route aggregation

- 2 methods
  - basic route redistribution: creates an aggregate route, then redistributes the routes in BGP
  - conditional aggregation: creates an aggregate route , then advertises or not certain routes based on route maps, AS-SET, or summary information
- **bgp suppress-inactive** stops BGP to advertise inactive routes (not installed into the RIB) to any peer.

BGP route aggregation generating AS_SET information

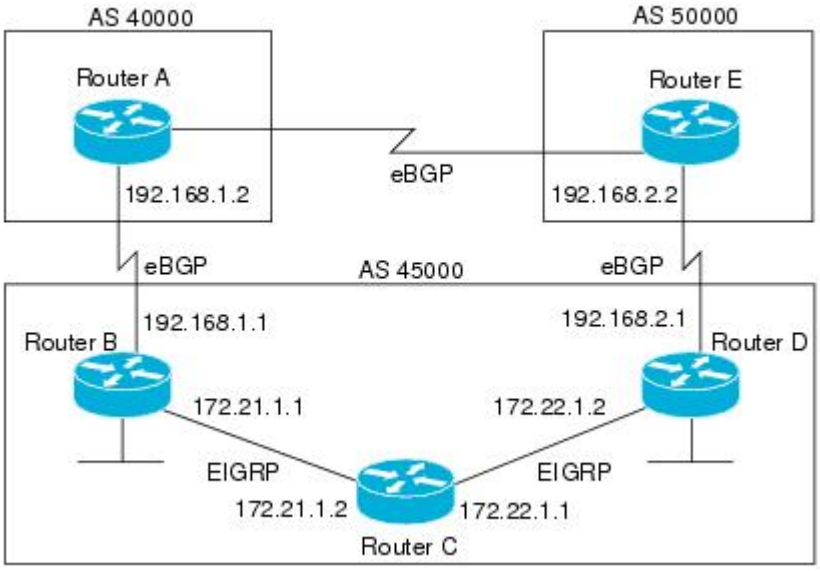
#TODO: improve this part

AS_SET information can be generated when BGP routes are aggregated using the aggregate-address

command. The path advertised for such a route is an AS_SET consisting of all the elements, including the communities, contained in all the paths that are being summarized. If the AS_PATHs to be aggregated are identical, only the AS_PATH is advertised. The ATOMIC-AGGREGATE attribute, set by default for the aggregate-address command, is not added to the AS_SET.

12.5.7. BGP backdoor routes

- Use **network backdoor** to cause BGP to prefer EIGRP



Task: Indicate a network reachable through a backdoor route

```
(config-router)# network <ip-address> backdoor
```

12.5.8. Best path selection algorithm

1. reachable next hop (well-known mandatory)
2. highest weight
3. highest local pref
4. locally originated paths (network, distribute, aggregate-summary) over externally originated paths
5. shortest AS path
6. lowest origin type (internal over external over incomplete)
7. lowest MED
8. eBGP paths over iBGP paths
9. lowest IGP cost
10. oldest path
11. lowest BGP router id

"We Love Oranges AS Oranges Mean Pure Refreshment". W Weight (Highest) L Local_Pref (Highest) O Originate (local originate) AS As_Path (shortest) O Origin Code (IGP < EGP < Incomplete) M MED (lowest) P Paths (External Paths preferred Over Internal) R Router ID (lowest)

community attributes

- No-advertise: prevents advertisements to any BGP peer
- No-export: prevents advertisements to any eBGP peer
- No-advertise: prevents advertisements outside the AS, or in confederation scenarios, outside the sub-AS
- Internet: advertises routes to any route

12.5.9. BGP Routing Process

Task: Configure a bgp routing process

```
(config)# router bgp <asn>
```

Task: Specify a network as local to the BGP routing table

```
(config-router)# network <prefix> [mask  <a.b.c.d>] [route-map <name>]
```

Task: Disable the IPv4 unicast address family for the BGP routing process

```
no bgp default ipv4-unicast
```

Task: Add a text description with a specified neighbor

```
(config-router)# neighbor <ip-address> description <text>
```

- Apply a route map to incoming or outgoing routes

```
(config-router)# neighbor <ip-address|peer-group-name> route-map <name> [in | out]
```

Aggregating Route Prefixes Using BGP

Task: Redistribute static routes into the BGP routing table

```
(config-router)# redistribute static
```

Task:Create an aggregate entry in a BGP routing table

```
(config-router)# aggregate-address <prefix> <mask> [as-set]
```

Task: Create an aggregate route and suppress advertisements of more-specific routes to all peers

```
(config-router)# aggregate-address <prefix> <mask> [summary-only]
```

Task: Create an aggregate route but suppress advertisement of specified routes

```
(config-router)# aggregate-address <prefix> <mask> [suppress-map <map-name>]
```

Task: Selectively advertises routes previously suppressed by the **aggregate-address** *command*

```
(config-router)# neighbor <ip-address | peer-group-name> unsuppress-map <map-name>
```

- Conditionally advertise BGP routes

The routes or prefixes that will be conditionally advertised are defined in two route maps: an advertise map and either an exist map or nonexist map. The route map associated with the exist map or nonexist map specifies the prefix that the BGP speaker will track. The route map associated with the advertise map specifies the prefix that will be advertised to the specified neighbor when the condition is met.

- If a prefix is found to be present in the exist map by the BGP speaker, the prefix specified by the advertise map is advertised.
- If a prefix is found not to be present in the nonexist map by the BGP speaker, the prefix specified by the advertise map is advertised.
- If the condition is not met, the route is withdrawn and conditional advertisement does not occur. All routes that may be dynamically advertised or not advertised must exist in the BGP routing table in order for conditional advertisement to occur. These routes are referenced from an access list or an IP prefix list.

Task: Advertise selectively some BGP routes to neighbor

```
(config-router)# neighbor <ip-address> advertise-map <name-1> { exist-map <name> |
non-exist-map <name>}
```

Task: Inject more specific prefixes into a BGP routing table over less specific prefixes

```
(config-router)# bgp inject-map <name> exist-map <name> [copy-attributes]
```

12.5.10. BGP Routes

Task: Advertise a default route to BGP peers

```
(config-router)# neighbor <ip-address> default-originate  [route-map <name>]
```

Task: Suppress inactive route advertisement using BGP

- Suppress inactive route advertisement

```
(config-router-af)# bgp suppress-inactive
```

12.5.11. peer session template

Task: Create a peer session template

```
(config-router)# template peer-session <name>
```

Task: Inherit the configuration of another peer session template

```
(config-router-stmp)# inherit peer-session <template-name>
```

Task: Send a peer session template to a neighbor so that the neighbor can inherit the configuration

```
(config-router)# neighbor <ip-address> inherit peer-session <template-name>
```

12.5.12. Peer Policy Template

Task: Create a peer policy template

```
(config-router)# template peer-policy <name>
```

Task: Configure the maximum number of prefixes that a neighbor will accept from this peer

```
(config-router-ptmp)# maximum-prefix <limit> [<threshold>] [restart <interval> |
warning-only]
```

A peer policy template can directly or indirectly inherit up to 8 peer policy templates.

A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies only from peer templates.

12.5.13. BGP Routing Table

Task: Display the entries in the bgp routing table

```
# sh ip bgp [prefix] [mask]
```

- Verify that the VRF instance has been created

```
# show ip vrf
```

- Display information about all the BGP paths in the database

```
# show ip bgp paths
```

- Display the status of all BGP connections

```
# show ip bgp summary
```

- Display IPv4 multicast database-related information

```
show ip bgp ipv4 multicast <command>
```

- Display injected paths

```
# show ip bgp injected-paths

BGP table version is 11, local router ID is 10.0.0.1
Status codes:s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes:i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       10.0.0.2                               0 ?
*> 172.17.0.0/16    10.0.0.2                               0 ?
```

- Display update replication stats for BGP update groups

```
# show ip bgp replication [<index-group> | <ip-address>] [summary]
```

- Display BGP routes that are not installed in the RIB

```
# show ip bgp rib-failure

Network            Next Hop                      RIB-failure   RIB-NH Matches
10.1.15.0/24       10.1.35.5          Higher admin distance              n/a
10.1.16.0/24       10.1.15.1          Higher admin distance              n/a
```

- Display locally configured peer session template

```
show ip bgp template peer-session
```

12.5.14. Troubleshoot

Task: Display info about the processing of BGP update groups.

```
# debug ip bgp groups
```

12.5.15. todos

- Concept: bgp route aggregation generating AS_SET information
- Multiprotocol bgp concepts
- Multiprotocol bgp extensions for IP multicast concepts
- AFI bgp address family identifier model : ipv4, ipv6,clns, vpnv4

12.6. Redistribution

- Redistribution occurs from the routing table not the routing database
- When redistributing protocol X into Y, take …
  - routes in the RIB via protocol X
  - connected interfaces running protocol X
- choose
  - routes with lower AD

12.6.1. Administrative distance

Route source	Distance
Connected route	0
Static route	1
summary EIGRP	5
eBGP	20

Route source	Distance
internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
ODR	160
External EIGRP	170
iBGP	200
Unknown	255

12.6.2. Spot issues

- Loops cannot occur with one single point of redistribution
- Loops may occur with multiple points of redistribution

12.6.3. heuristics

- identify each domain and associate a tag
- assign tags to each domain

```
route-map ospf2eigrp permit
  set tag 123
```

- deny tag on re-entry
  - always block routes to be re-enter the domain
  - optionaly: block routes as per scenario requirement

```
! block own routes
route-map ospf2eigrp deny 10
  match tag 456
! block some routes if requested
route-map deny 20
  match tag 78
```

- all tags to pass through transit domains without re-tagging them

```
! identify the transit tags without tagging
route-map ospf2eigrp permit 60
  match tag 234
```

- Use BGP community instead of tags for BGP

```
route-map ospf2bgp permit 70
  set community 110
```

```
ip community-list 1 permit 10
  match community 1
  set tag 110
```

12.6.4. Connected routes

Task: redistribute connected routes

```
redistribute connected
```

- OVerride the implicit redistribution of interfaces running the protocol X

12.6.5. Static routes

Task: redistribute static route

```
(config-router)# redistribute static route-map <name> metric <value>
(config-router)# default-metric <hops>
```

12.6.6. RIP

- doesn't differentiate between internal and external routes
- no default seed metric
  - recommendation: use 1 as default-metric

Task: Prevent loss of packet when BGP routes are redistributed in RIP

```
(config)# router rip
(config-router)# input-queue 1024
```

12.6.7. EIGRP

```
redistribute <protocol> metric <bandwidth> <delay> <load> <reliability> <MTU>
```

- internal routes AD < external routes
- uses router-id for loop prevention
- no default seed metric unless EIGRP to EIGRP
  - default-metric <bandwidth> <delay> <load> <reliability> <MTU>
  - default-metric 10000 100 1 255 1500

Duplicate router-ids will prevent EIGRP to install routes

12.6.8. OSPF redistribution

- differentiates between internal and external routes but same AD= 110
- Router-id for flooding loop prevention
- Use **subnets** keyword
- default metric is 1 for BGP and 20 for other IGP
- default metric-type E2/N2
- OSPF path selection TODO: improve this part
  - E1 > E2 > N1 > N2
  - E1 & N1 vs E2 & N2 metrics

```
router ospf 1
  redistribute rip
  redistribute eigrp
  default-metric 10
```

Task: Assign different AD to internal and external

```

```

12.6.9. BGP redistribution

IGP to BGP

- denies OSPF external routes by default

Task: redistribute OSPF into BGP

```
redistribute ospf <pid> match internal external
```

BGP to IGP

- iBGP routes denied by default, eBGP routes win

Part III : VPN Technologies

Chapter 13. MPLS

13.1. Concepts

- mpls vpn
  - ce : no mpls-aware
  - pe : mpls and vpn aware
  - p : no vpn aware

13.1.1. MPLS label stack

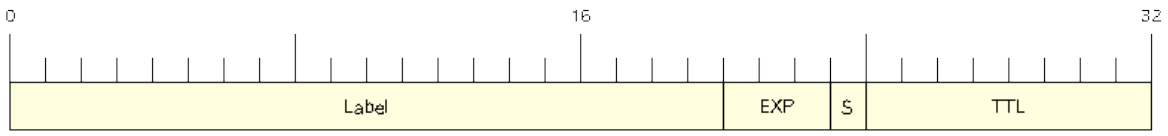


Figure 29. MPLS header format

Label
: Locally significant to the router

EXP
: Experimental, class of service

S
: Bottom-of-Stack flag

TTL
: Time to live

13.1.2. label distribution

- protocol : ldp (default rfc 3036) or tdp (cisco)

dynamic discovery of adjacent ldp peers

- neigbhbor discovery UDP port 646
  - basic neighbor discovery: multicast hellos to directly connected neighbors
  - extended neighbor discovery: unicast hellos to non-directly connected neighbors

```
R1#sh mpls ldp neighbor
  Peer LDP Ident: 192.1.5.5:0; Local LDP Ident 192.1.1.1:0
    TCP connection: 192.1.5.5.21288 - 192.1.1.1.646
    State: Oper; Msgs sent/rcvd: 10/11; Downstream
    Up time: 00:04:24
    LDP discovery sources:
      FastEthernet0/0, Src IP addr: 172.16.15.5
    Addresses bound to peer LDP Ident:
    172.16.15.5 192.1.5.5
```

- timers: hello interval (5 seconds) and holdtime (15 seconds)

```
(config)# mpls ldp discovery hello interval <sec>
(config)# mpls ldp discovery hello holdtime <sec>
```

peering establishment

in 2 steps:

- transport connection: if the 2 peers have never established a tcp session, create a new session with a client (active device, highest ip address) using a random port and the server (lowest ip addr) listening on the TCP 646 port.

```
R1#show tcp brief (state)
TCB        Local_Address   Foreign_Address
498D80D8   192.1.1.1.646   192.1.5.5.21288   ESTAB
```

- session establishment:
  - negotiates ldp protocol version, label exchange method, timers
  - if incompatibility, sends error negotiation messages and restart the negotiation with initial backoff value (15 seconds) and maximum backoff value (120 seconds)

```
R1#show mpls ldp parameters
  Protocol version: 1
  Session hold time: 180 sec; keep alive interval: 60 sec
  Discovery hello: holdtime: 15 sec; interval: 5 sec
  Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
  Downstream on Demand max hop count: 255
  Downstream on Demand Path Vector Limit: 255
  LDP for targeted sessions
  LDP initial/maximum backoff: 15/120 sec
  LDP loop detection: off
```

13.1.3. regulation of peer-to-peer communication and label exchange

- with keepalives (60 seconds)

```
(config)# mpls ldp holdtime <sec>
%Previously established sessions may not use the new holdtime.
```

- keepalive timer is reset every time ldp packets or keepalive are received
- keepalive are automatically adjusted to 1/3 of the configured holdtime
- you need to reset the tcp session for new timers to take effect

13.2. label distribution and control

- methods:
  - Unsolicited downstream distribution mode
  - solicited downstream distribution mode

(to be revised)

13.2.1. Route distinguishers and route targets

RD and RT

RD is added to prefix when BGP vpnv4 advertised NLRI. - only one per NLRI RT is Path attribute of the NLRI - 1 or + for each RD/prefix - useful in overlapping VPNs or central service VPN offered by SP

13.3. Commands

```
show mpls forwarding-table
```

check http://www.cisco.com/en/US/docs/ios-xml/ios/mpls/command/mp-s2.html#wp4232274342

Chapter 14. GRE

- tunnelling protocol developed by Cisco
- IP protocol 47
- RFC 2784
- RFC 2345
- RFC 1234

14.1. Tunneling

- Tunneling uses encapsulates data packets with from one protocol inside a different protocol and transports the data packets unchanged across a foreign network.
- Unlike encapsulation, tunneling allows a lower-layer protocol, or same-layer protocol, to be carried through the tunnel.
- **Passenger protocol** : The protocol that you are encapsulating. Examples: AppleTalk, IP, IPX.
- **Carrier protocol** : The protocol that does the encapsulating. Examples: GRE, IP-in-IP, L2TP,MPLS, STUN,DLSw+.
- **Transport protocol** : The protocol used to carry the encapsulated protocol. The main transport protocol is IP.

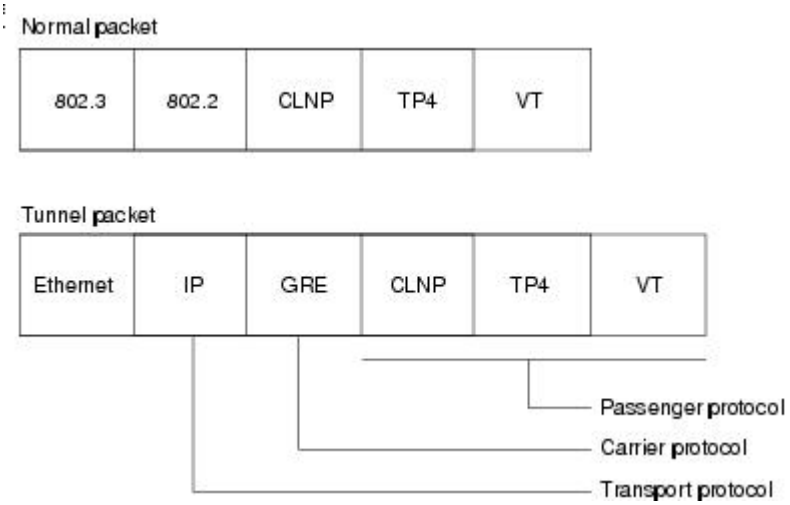
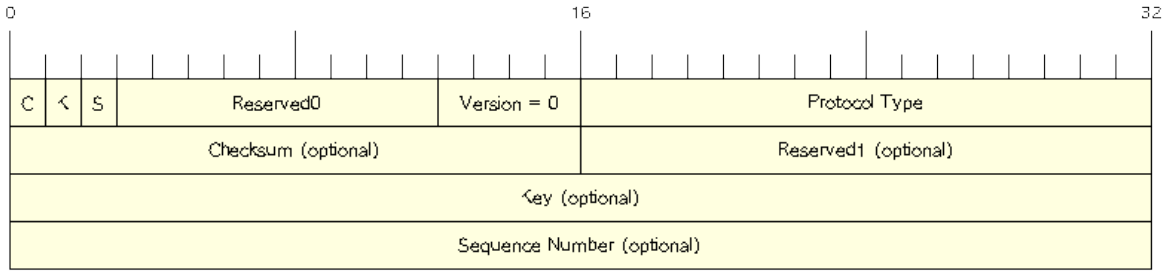


Figure 30. IP Tunneling Terminology and Concepts

14.2. GRE header



C

- Checksum present
- Set to 1 if the checksum and the Reserved1 field are present

K

- Key bit. Set to 1 if

Reserved0

- if any of bits 1-5 are non-zero, a receiver must discard the packet unless receiver implements RFC1701

Protocol Type

- Ether protocol type (e.g. IPv4 → 0x800)

14.2.1. GRE keepalive

The GRE tunnel keepalive mechanism gives the ability for one side to originate and receive keepalive packets to and from a remote router even if the remote router does not support GRE keepalives. For GRE keepalives, the sender pre-builds the keepalive response packet inside the original keepalive request packet so that the remote end only needs to do standard GRE decapsulation of the outer GRE IP header and then forward the inner IP GRE packet. GRE tunnel keepalives timers on each side are independent and do not have to match. The problem with the configuration of keepalives only on one side of the tunnel is that only the router that has keepalives configured marks its tunnel interface as down if the keepalive timer expires. The GRE tunnel interface on the other side, where keepalives are not configured, remains up even if the other side of the tunnel is down. The tunnel can become a black-hole for packets directed into the tunnel from the side that did not have keepalives configured.

Specifies the number of times that the device will continue to send keepalive packets without response before bringing the tunnel interface protocol down.

GRE keepalive packets may be configured either on only one side of the tunnel or on both. If GRE keepalive is configured on both sides of the tunnel, the period and retries arguments can be different at each side of the link.

This command is supported only on GRE point-to-point tunnels.

Task: Configure GRE keepalives

```
(config-if)# keepalive [ period [retries]]
```

14.3. GRE tunnel

To build a tunnel, a tunnel interface must be defined on each of two routers and the tunnel interfaces must reference each other. At each router, the tunnel interface must be configured with a L3 address. The tunnel endpoints, tunnel source, and tunnel destination must be defined, and the type of tunnel must be selected.

Optional steps can be performed to customize the tunnel.

Remember to configure the router at each end of the tunnel. If only one side of a tunnel is configured, the tunnel interface may still come up and stay up (unless keepalive is configured), but packets going into the tunnel will be dropped.

Task: Summary steps

```
interface tunnel number
  bandwidth kbps
  keepalive [period [retries]]
  tunnel source {ip-address | interface-type interface-number}
  tunnel destination {hostname | ip-address}
  tunnel key key-number
  tunnel mode [gre {ip | multipoint} | dvmrp | ipip | mpls | nos]
  ip mtu bytes
  ip tcp mss mss-value
  tunnel path-mtu-discovery [age-timer {aging-mins| infinite}]
```

The router should have a route to the destination address, but not through the tunnel interface.

14.3.1. Configuration example

Note that Ethernet interface 0/1 is the tunnel source for Router A and the tunnel destination for Router B. Fast Ethernet interface 0/1 is the tunnel source for Router B and the tunnel destination for Router A.

Router A

```
interface Tunnel0
 ip address 10.1.1.2 255.255.255.0
 tunnel source Ethernet0/1
 tunnel destination 192.168.3.2
 tunnel mode gre ip
!
interface Ethernet0/1
 ip address 192.168.4.2 255.255.255.0
```

Router B

```
interface Tunnel0
 ip address 10.1.1.1 255.255.255.0
 tunnel source FastEthernet0/1
 tunnel destination 192.168.4.2
 tunnel mode gre ip
!
interface FastEthernet0/1
 ip address 192.168.3.2 255.255.255.0
```

14.4. Troubleshooting

Three reasons for a GRE tunnel to shut down:

- There is no route to the tunnel destination address.
- The interface that anchors the tunnel source is down.
- The route to the tunnel destination address is through the tunnel itself. “%TUN-5-RECURDOWN:Tunnel0“

With the above three reasons for tunnel shut down are problems local to the router at the tunnel endpoints and do not cover problems in the intervening network.

Also if the two routers tunnel modes do not match, the tunnel interface can still stay in an up/ip state but the routers cannot forward packets because of the mismatch encapsulation.

14.4.1. "%TUN-5-RECURDOWN" error message and flapping EIGRP/OSPF/BGP neighbors over a GRE tunnel

http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/22327-gre-flap.html

14.5. Questions

1. What is the minimum amount of additional header that GRE adds to a packet?
    a. 16 bytes

20 bytes

c. 24 bytes

d. 36 bytes

e. 48 bytes

2. Which of the following are valid options in a GRE header (select all that apply)?

   a. GRE Header Length

   b. Checksum Present

   c. Key Present

   d. External Encryption

   e. Protocol

3. What is the purpose of a GRE tunnel interface?

   a. It is always the tunnel source interface.

   b. It is always the tunnel destination interface.

   c. It is where the protocol that travels through the tunnel is configured.

   d. It is the interface that maps to the physical tunnel port.

   e. It is not used today

http://ptgmedia.pearsoncmg.com/9781587201509/samplechapter/158720150X_CH14.pdf

b.

Chapter 15. DMVPN

15.1. Concepts

- dynamic creation of spoke-to-spoke
- DMVPN = mGRE + NHRP + IPSec

Start with mGRE configuration

interface tunnel 0 ip address 141.11.10.1 255.255.255.0 tunnel source e0 tunnel mode gre multipoint

15.2. Phases

Phase 1

- Hub and spoke functionality
- for simplified and smaller configuration
- zero touch provisioning for adding spokes to the VPN
- supports dynamically addressed CPEs

Phase 2

- Spoke-to-spoke functionality
- on demand spoke-to-spoke tunnels avoids dual encrypts/decrypts
- Smaller spoke CPE can participate in the virtual mesh

Phase 3

Architecture and scaling

Task: Verify NHRP configuration

```
# sh ip nhrp

141.11.10.2/32 via 141.11.10.2
   Tunnel1234 created 00:02:21, expire 00:57:38
   Type: dynamic, Flags: unique registered
   NBMA address: 10.11.10.2
141.11.10.3/32 via 141.11.10.3
   Tunnel1234 created 00:02:09, expire 00:57:50
   Type: dynamic, Flags: unique registered
   NBMA address: 10.11.10.3
```

Chapter 16. IPSEC

Part IV : Infrastructure Security

Chapter 17. Device security

17.1. AAA

17.1.1. PPP security

- implement device access control
  - lines
  - password encryption
  - management plane protection
  - control plane protection

Chapter 18. Network security

18.1. Switch security

18.2. Access control list

18.2.1. References

sec-data-acl

18.2.2. Reflexive ACL

- sourcecontrol
- dhcp snooping
- ip source-guard
- dynamic arp inspection
- port-security

18.3. Router security

- ipv4 access list
- ipv6 traffic filter
- unicast reverse path forwarding
- ipv6 first hop security
  - ra guard
  - dhcp guard
  - binding table
  - device tracking
  - ND inspection/snooping
  - source guard
  - PACL

Chapter 19. IEEE 802.1X

19.1. Definition

- Port-based authentication
- until the client is authenticated, allows only EAPoL (Extensible Authentication Protocol over LAN), CDP and STP traffic
- supplicant: client workstation running 802.1x compliant software
- authenticator: edge switch or wireless access point
- authentication server: performs the actual authentication (Radius with EAP)

19.2. Port security

Part V : Infrastructure Services

Chapter 20. System management

20.1. Telnet

Task: Description

```
(config)# ip telnet hidden addresses
```

20.2. SSH

blabla xxxx

20.3. SCP

20.4. [T]FTP

20.4.1. FTP client

TODO

20.4.2. FTP server

TODO

20.5. SNMP

Configuration guides | **Network Management** | **configuring SNMP support**

- Application-layer protocol between SNMP managers and agents
  - SNMP manager running NMS software:i UDP port 162 open for traps/informs messages
  - SNMP managed devices running SNMP agent: UDP 161 open for GET/SET messages

20.5.1. Version

v1
: original specs, weak authentication with community string, Uses SMIv1, uses MIB-I originally.

v2
: Uses SMIv2, removed requirement for communities, added GetBulk and Inform messages, but began with MIB-II originally.

v2c
: 64-bit counters, getBulkRequest, informsRequest,Pseudo-release (RFC 1905) that allowed SNMPv1-style communities with SNMPv2;

v3

authentication, encryption, Mostly identical to SNMPv2, but adds significantly better security, although it supports communities for backward compatibility. Uses MIB-II.

20.5.2. MIB

MIB: dictionaries of OID

OID: hierarchical identifiers in numerical format that represent MIB variables. (e.g. 1.3.6.1.2.1 "Interfaces" 1.3.6.1.4.1.9 "Enterprises - Cisco")

20.5.3. Packet format

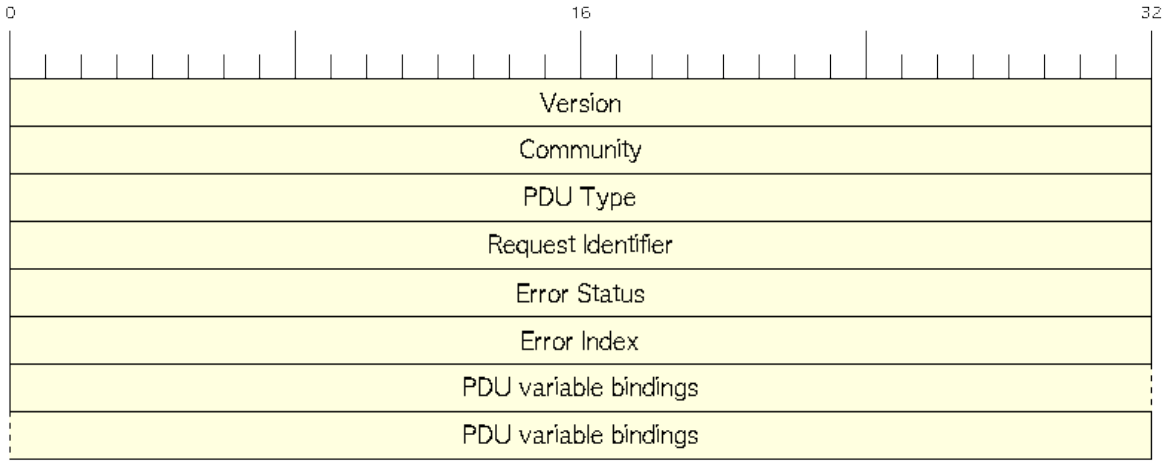


Figure 31. SNMP header format

SNMP PDU Types

1. SetRequest
2. GetRequest
3. GetNextRequest
4. GetBulkRequest
5. Trap
6. InformsRquest
7. Response

20.5.4. Basic system information

Task: Configure the location information

```
(config)# snmp-server location <homesweethome>
```

Task: Configure the contact information

```
(config)# snmp-server contact <no-where-to-be-found>
```

Task: Configure the system serial number

```
(config)# snmp-server chassis-id <system-serial-number>
```

20.5.5. Views

Task: Create a view

```
(config)# snmp-server view <name> <oid-tree> {included | excluded}
```

20.5.6. Communities

Task: Configure a community string

```
(config)# snmp-server community <string> [view <name>] [ro|rw] [<acl]
```

Task: Display community string

```
# sh snmp community
```

20.5.7. Traps

Task: Send traps to NMS

```
(config)# snmp-server host <ip-address> [traps | informs] [version {1| 2c | 3 [auth |
noauth | priv]}] community-string [udp-port port-number] [notification-type]
```

20.5.8. SNMP v3

Task: Configure SNMP v3 group

```
(config)# snmp-server group [<groupname> {v1 | v2c | v3 [auth | noauth | priv]}]
            [read <readview>]
            [write <writeview>]
            [notify <notifyview>]
            [access <acl]
```

Task: Display SNMP v3 group settings

```
# sh snmp group
```

Task: Configure SNMP v3 user

```
(config)# snmp-server engineID {local <engine-id> | remote <ip-address> [udp-port
<number> ] [vrf <vrf-name> ] <engine-id-string> }
(config)# snmp-server user <username> <groupname> [remote <ip-address> [udp-port
<number> ]] {v1 | v2c | v3 [encrypted] [auth {md5 | sha} <auth-password> ]} [access
<acl>]
```

Task: Display SNMP user information

```
# sh snmp user <user>
```

Task: Display SNMP engineID

```
# sh snmp engineID
```

20.5.9. SNMP manager

Task: Configure the SNMP manager process

```
(config)# snmp-server manager
```

Task: Configure the SNMP manager session time-out

```
(config)# snmp-server manager session-timeout <seconds>
```

Task: Display the status of the SNMP sessions

```
# sh snmp sessions brief
```

Task: Display the current set of pending SNMP requests

```
# sh snmp pending
```

20.5.10. SNMP shutdown mechanism

Task: Enable the SNMP shutdown mechanism

```
(config)# snmp-server system-shutdown
```

Task: Define the maximum SNMP agent packet size

```
(config)# snmp-server packetsize <bytes>
```

Task: Specify the TFTP servers used for saving and loading configuration files

```
(config)# snmp-server tftp-server-list <acl>
```

Task: Disable SNMP agent

```
(config)# no snmp-server
```

20.6. RMON

TODO clean this section

Remote Monitoring, or RMON, is an event-notification extension of the SNMP capabil- ity on a Cisco router or switch. RMON enables you to configure thresholds for alerting based on SNMP objects, so that you can monitor device performance and take appropri- ate action to any deviations from the normal range of performance indications. RMON is divided into two classes: alarms and events. An event is a numbered, user- configured threshold for a particular SNMP object. You configure events to track, for example, CPU utilization or errors on a particular interface, or anything else you can do with an SNMP object.

You set the rising and falling thresholds for these events, and then tell RMON which RMON alarm to trigger when those rising or falling thresholds are crossed. For example, you might want to have the router watch CPU utilization and trigger an SNMP trap or log an event when the CPU utilization rises faster than, say, 20 percent per minute. Or you might configure it to trigger an alarm when the CPU utiliza- tion rises to some absolute level, such as 80 percent. Both types of thresholds (relative, or delta," and absolute) are supported. Then, you can configure a different alarm notifica- tion as the CPU utilization falls, again at some delta or to an absolute level you specify.

The alarm that corresponds to each event is also configurable in terms of what it does (logs the event or sends a trap). If you configure an RMON alarm to send a trap, you also need to supply the SNMP community string for the SNMP server.

Event and alarm numbering are locally significant. Alarm numbering provides a pointer to the corresponding event. That is, the configured events each point to specific alarm num- bers, which you must also define.

Example

```
rmon event 1 log trap public description Fa0.0RisingErrors owner config
rmon event 2 log trap public description Fa0.0FallingErrors owner config
rmon event 3 log trap public description Se0.0RisingErrors owner config
rmon event 4 log trap public description Se0.0FallingErrors owner config
rmon alarm 11 ifInErrors.1 60 delta rising-threshold 10 1 falling-threshold 5 2 owner
config
rmon alarm 20 ifInErrors.2 60 absolute rising-threshold 20 3 falling-threshold 10 4
owner   config
```

Task: Monitor RMon activity

```
# sh rmon activity
# sh rmon event
```

20.7. Syslog

- RFC 5424
- Clear-text protocol that provides event notifications without requiring difficult, time-intensive configuration or opening attack vectors.

Steps

- Install a Syslog server on a workstation with a fixed IP address.
- Configure the logging process to send events to the Syslog server's IP address using the **logging host** command.
- Configure any options, such as which severity levels (0–7) you want to send to the Syslog server using the logging trap command.

20.8. NTP

Configuration guides | Network Management | Basic System Management | Setting Time and Calendar Services

- Version 3 RFC 1305
- UDP port 123
- IOS does not support stratum 1 service, cannot be linked to stratum 0 atomic clock
- Accuracy < milliseconds with 1 NTP packet per minute
- Stratum
  - 0 : atomic clock
  - 8 : default value

20.8.1. NTP associations

- Polled-based : better accuracy and reliability
  - Client mode : **ntp server**
  - Symmetric active mode : **ntp peer**
- Broadcast-based : less manual configuration on LAN
  - Server : **ntp broadcast**
  - Client : **ntp broadcast client**

Task: Verify NTP status

```
# show ntp status
```

Task: Verify NTP associations

```
# show ntp associations
```

Task: Troubleshoot NTP associations

```
# debug ntp refclock
```

Polled-based associations

```
(config)# ntp server <ip-address> [normal-sync] [version <number>] [key <id>] [prefer]
(config)# ntp peer <ip-address> [normal-sync] [version <number>] [key <id>] [prefer]
```

Broadcast-based associations

```
(config-if)# ntp broadcast version <number>
(config-if)# ntp broadcast client
(config-if)# ntp broadcastdelay <microseconds>
```

20.8.2. NTP access groups

Task: Grant/deny access privileges with ipv4 or ipv6 access-lists

```
(config)# ntp access-group [ipv4 | ipv6] <options>  <access-list-id> [kod]
```

- *options* per increasing order of restrictions are:
  - **peer**: synchronize itself to systems whose address passes access list criteria
  - **serve**: allows time requests and NTP control queries but no synchronization
  - **serve-only**: allows only time requests
  - **query-only**: allows only NTP control queries
- **kod** sends the kiss-of-death packet to any host that tries to send a packet that is not compliant with the access-group policy.

20.8.3. NTP authentication

- Use cryptographic checksum keys
- Encryption/decryption are CPU-intensive and may degrade accuracy

```
(config)# ntp authenticate
(config)# ntp authentication-key <number> md5 <key>
(config)# ntp trusted-key <key-number> [-<end-key-number]
(config)# ntp server <ip-address> key <id>
```

20.8.4. Source IP address

- Default to NTP packet outgoing interface

Task: Change the source IP address for all destinations

```
(config)# ntp source interface
```

Task: Change the source for a specific association

```
(config)# ntp {server|peer} source
```

20.8.5. Authoritative server

Task:

```
(config)# ntp master
```

20.8.6. Panic treshold

Task: Reject time updates greater than the panic threshold of 1000 seconds

```
(config)# ntp panic update
```

20.8.7. Orphan mode

- When a subnet lost communications with clock servers
- Orphan parent simulate a UTC source for orphan children

```
(config)# ntp server <a.b.c.d>
(config)# ntp peer <e.f.g.h>
(config)# ntp orphan <stratum>
```

20.8.8. external reference clock

```
# line aux <number>
# ntp refclock trimble pps none stratum <number>
```

20.8.9. Software clock

```
(config)# clock timezone <zone> <hours-offset> [<minutes-offset>]
(config)# summer-time <zone> recurring [<week day month hh:mm> [<offset>]]
(config)# summer-time <zone> date [<date month year hh:mm> [<offset>]]
# clock set <hh:mm:ss date month year>
# show clock
```

20.8.10. Hardware-clock

- different from software-clock

```
# calendar set <hh:mm:ss date month year>
(config)# clock calendar-valid
# clock read-calendar
# clock update-calendar
# show calendar
# show clock [detail]
# show ntp associations [details]
# show ntp status
```

20.8.11. Time Ranges

Task: Configure time ranges

```
(config)# time-range <name>
(config-time-range)# absolute [start <hh:mm date month year>] [end <hh:mm date month
year>]
(config-time-range)# periodic <day-of-week> <hh:mm> to [<day-of-the-week>] <hh:mm>
```

Task: Verify time range

```
# show time-range
```

20.8.12. Vulnerability

- DoS for version ⇐ 4.2.4p7
- No workaround, disable NTP on the device

20.9. HHTP

Cisco IOS routers and switches support web access for administration, through both HTTP and HTTPS. Enabling HTTP access requires the ip http server global configura- tion command. HTTP access defaults to TCP port 80. You can change the port used for HTTP by configuring the ip http port command. You can restrict HTTP access to a rout- er using the ip http access-class command, which applies an extended access list to con- nection requests. You can also specify a unique username and password for HTTP access using the ip http client username and ip http client password commands. If you choose, you can also configure HTTP access to use a variety of other access-control methods, including AAA, using ip http authentication [aaa | local | enable | tacacs].

You can also configure a Cisco IOS router or switch for Secure Sockets Layer (SSL) access. By default, HTTPS uses TCP port 443, and the port is configurable in much the same way as it is with HTTP access. Enabling HTTPS access requires the ip http secure- server command. When you configure HTTPS access in most IOS Release 12.4 ver- sions, the router or switch automatically disables HTTP access, if it has been configured. However, you should disable it manually if the router does not do it for you.

HTTPS router access also gives you the option of specifying the cipher suite of your choice. This is the combination of encryption methods that the router will enable for HTTPS access. By default, all methods are enabled

Task: Display HTTP secure server status

```
# sh ip http server secure status

HTTP secure server status                : Enabled
HTTP secure server port                  : 443
HTTP secure server ciphersuite           : 3des-ede-cbc-sha des-cbc-sha rc4-128-md5
rc4-128-sha
HTTP secure server client authentication  : Disabled
HTTP secure server trustpoint            :
HTTP secure server active session modules : ALL
```

Chapter 21. QoS

21.1. MQC

21.1.1. MQC structure

- define a traffic class with **class map**
- create a traffic policy with **policy map**
- attach the traffic policy to an interface with **service-policy**
- restrictions: maximum of 256 classes in a single policy map

21.1.2. Fields that can be marked for QoS purposes

- IPP, IP DSCP, DS field, ToS byte, CoS,Frame Relay DE, ATM CLP, MPLS EXP

IPP

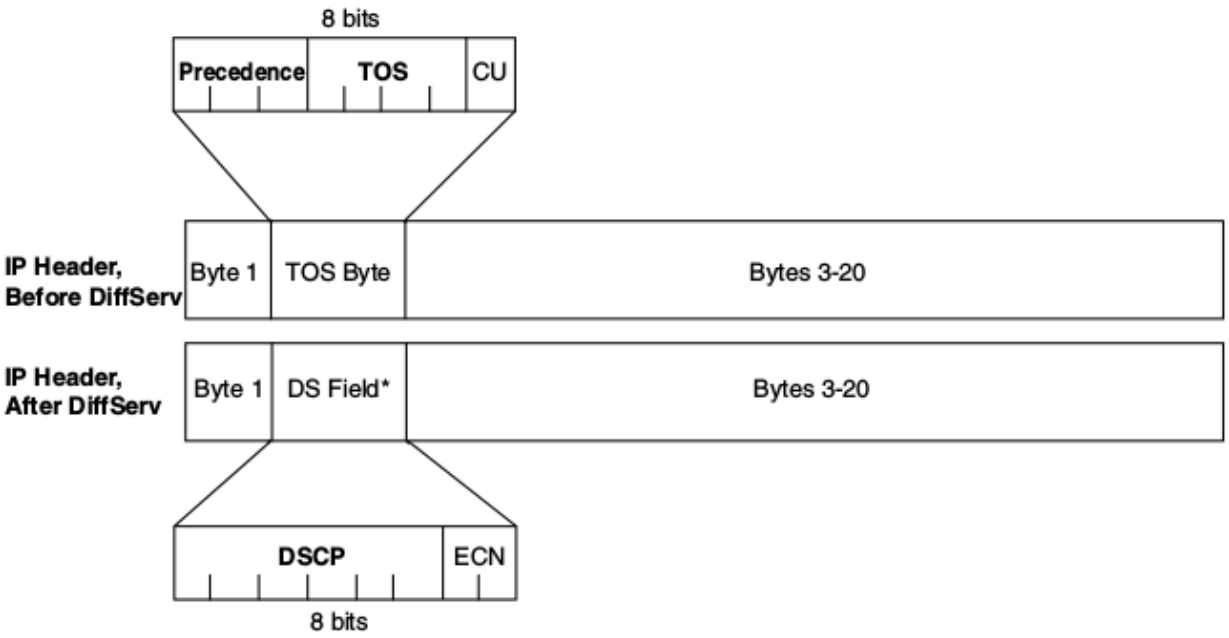


Table 12. IPP and Class Selector DSCP values

DSCP CS name	IPP names
CS0/Default	routine
CS1	priority
CS2	immediate
CS3	flash
CS4	flash override
CS5	critical
CS6	internetwork control

CS7	network control

DSCP

- PHB: Per-hop Behavior
- Expedited Forwarding:
- packets given queunino

21.1.3. Elements of a traffic class

A traffic class contains 3 major elts: a name, a series of **match** commands followed by **match-all** (default) or **match-any**

match commands that can be used with MQC

- match access-group
- match any
- match class-map
- match cos
- match destination-address mac
- match discard-class
- match [ip] dscp
- match field
- match fr-dlci
- match input-interface
- match ip rtp
- match mpls experimental
- match mpls experimental topmost
- match not // add a note here
- match packet length -
- etc

Command	Purpose
match protocol	Configures the match criteria for a class map on the basis of the specified protocol.
match protocol citrix	Configures NBAR to match Citrix traffic.
match protocol fasttrack	Configures NBAR to match FastTrack peer-to-peer traffic.
match protocol gnutella	Configures NBAR to match Gnutella peer-to-peer traffic.

Command	Purpose
match protocol http	Configures NBAR to match HTTP traffic by URL, host,MIME type, or fields in HTTP packet headers.
match protocol rtp	Configures NBAR to match Real-Time Transport Protocol (RTP) traffic.
match qos-group	Identifies a specific QoS group value as a match criterion.
match source-address mac	Uses the source MAC address as a match criterion.
match start	Configures the match criteria for a class map on the basis of the datagram header (Layer 2) or the network header (Layer 3).
match tag	Specifies tag type as a match criterion.

nbar cannot classify ipx nbar cannot classify multicast traffic

21.1.4. Elements of a traffic policy

- contains 3 elts: name, traffic class, command to enable the QoS feature

Command	Purpose
random-detect discard-class	Configures the WRED parameters for a discard- class value for a class in a policy map.
random-detect discard-class-based	Configures WRED on the basis of the discard class value of a packet.
random-detect ecn	Enables explicit congestion notification (ECN).
random-detect exponential-weighting-constant	Configures the exponential weight factor for the average queue size calculation for the queue reserved for a class.
random-detect precedence	Configure the WRED parameters for a particular IP Precedence for a class policy in a policy map.
set atm-clp	Sets the cell loss priority (CLP) bit when a policy map is configured.
set cos	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
set discard-class	Marks a packet with a discard-class value.
set [ip] dscp	Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte.
set fr-de	Changes the discard eligible (DE) bit setting in the address field of a Frame Relay frame to 1 for all traffic leaving an interface.
set mpls experimental	Designates the value to which the MPLS bits are set if the packets match the specified policy map.
set precedence	Sets the precedence value in the packet header.
set qos-group	Sets a QoS group identifier (ID) that can be used later to classify packets.

Command	Purpose
shape	Shapes traffic to the indicated bit rate according to the algorithm specified.
shape adaptive	Configures a Frame Relay interface or a point-to- point subinterface to estimate the available bandwidth by backward explicit congestion notification (BECN) integration while trafficommand
shape fecn-adaptative	configures a Frame Relay interface to reflect FECN bits as BECN bits in Q.922 test response messages.

Service policy

```
(config-if)# service-policy {input | output } policy-map-name
```

21.1.5. Verification

Display all class maps and their matching criteria

```
> show class-map
```

Display the configuration for the specified class of the specified policy map

```
> show policy-map policy-name class class-name
```

Display the configuration of all classes for all existing policy maps

```
> show policy-map interface interface-name interface-number
```

21.2. RSVP

21.2.1. Concepts

21.2.2. Configuration tasks

```
ip rsvp bandwidth [interface-kbps [single-flow-kbps] ]
```

- default 75 % can be allocated by to RSVP and up to 75% of the total bandwidth is available for a single flow

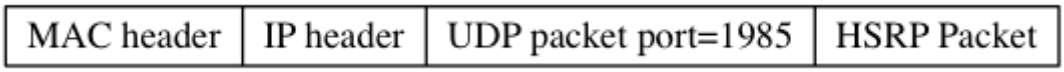
Chapter 22. First Host Redundancy Protocols

22.1. HSRP

Configuration Guides | First Hop Redundancy Protocols | HSRP Version 2

- RFC 2281
- Set of routers sharing one virtual IP address and one virtual MAC address
- Elect one active router with the highest (priority, IP address)
- Standy router takes over if Active router fails
- Elect new standby router if standby router fails or becomes the active router.
- To minimize network traffic, only the active and standby router send periodic HSRP messages.
- A router may participate in multiple groups with seperate state and timers for each group
- Unique group id per vlan
- Multicast address = 0000.0c07.AC*xy* (where xy is the HSRP group number in Hex)
- Send to multicast 224.0.0.2 for v1 and 224.0.0.102 for v2
- UDP port 1985

22.1.1. HSRP packet



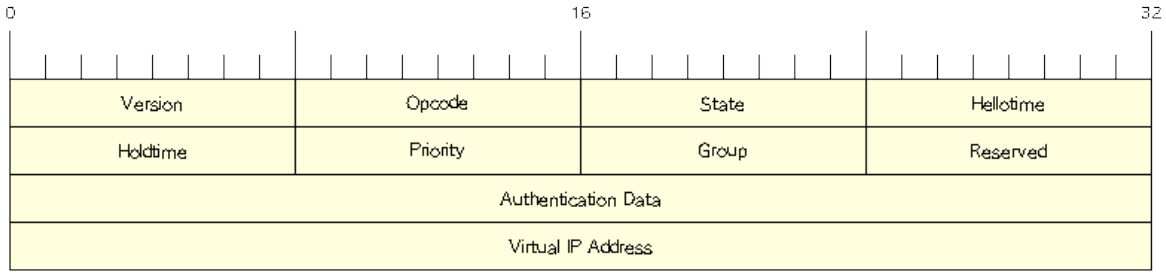


Figure 32. HSRP packet

22.1.2. HSRP version

- Version 1 by default
- Version 2 doesn't interoperate with HSRPv1 on the same interface.
  - v1 and v2 can run on different physical interfaces of the same router
  - v1 cannot advertise and learn millisecond timers
  - v1 uses multicast 224.0.0.2 and v2 224.0.0.102
  - v2 uses jitter timers (negative for hellotimes and positive for holdtimes)

- Group number: 0..255 for v1 and 0..4095 for v2
- Different virtual MAC address

Task: Change HSRP version

```
(config-if)# standby version {1 | 2}
```

22.1.3. HSRP OpCode

- Uses the Opcode field for preemption
  - 0: **Hello**: The router is running and is capable of becoming the active or standby router
  - 1: **Coup**: The router wishes to become the active router
  - 2: **Resign**: The router no longer wishes to be active router
- No preemption by default
- Preemption: the router with the highest priority becomes immediately the active router by sending a **coup** message, The previous active router changes to the **speak** state and sends a **resign** message.
- Can specify a delay before take over to allow the router to populate its routing table
  - **minimum** seconds after the last restart:
  - **reload** seconds after the first interface-up event after the device has reloaded, if such an event occurs within 360 seconds from reload. ???
  - **sync** seconds: for IP redundancy clients only ???

Task: Configure HSRP preemption

```
(config-if)# standby [ <group-number> ] preempt [ delay{ [ minimum <seconds> ] [
reload <seconds> ] [ sync <seconds> ] } ]
```

22.1.4. HSRP State

Code	State	Description
0	Initial	This is the starting state and indicates that HSRP is not running. This state is entered via a configuration change or when an interface first comes up.
1	Learn	The router has not determined the virtual IP address, and not yet seen an authenticated Hello message from the active router. In this state the router is still waiting to hear from the active router.
2	Listen	The router knows the virtual IP address, but is neither the active router nor the standby router. It listens for Hello messages from those routers.

Code	State	Description
4	Speak	The router sends periodic Hello messages and is actively participating in the election of the active and/or standby router. A router cannot enter Speak state unless it has the virtual IP address.
8	Standby	The router is a candidate to become the next active router and sends periodic Hello messages. Excluding transient conditions, there MUST be at most one router in the group in Standby state.
16	Active	The router is currently forwarding packets that are sent to the group's virtual MAC address. The router sends periodic Hello messages. Excluding transient conditions, there MUST be at most one router in Active state in the group.

22.1.5. Priority

- Default value: 100
- The higher (priority || IP address) wins

Task: Configure HSRP priority

```
(config-if)# standby [group-number] priority <number>
```

22.1.6. HSRP timers

Hellotime

- 3 seconds by default
- Only meaningful in Hello messages
- Configured on the router or learned from authenticated Hello message from the active router
  - not learned if HSRP hellos < 1 second

Holdtime

- 10 seconds by default
- >= 3 * hellotime

Task: configure HSRP timers

```
(config-if)# standby [group-number] timers[msec] <hellotime> [msec] <holdtime>
```

22.1.7. HSRP authentication

- Clear-text or MD5 encryption

Task: Configure HSRP clear-text authentication

```
(config-if)# standby [group-number] authentication text <string>
```

Task: Configure HSRP MD5 authentication

```
(config-if)# standby [group-number] authentication  md5 { key-string [ 0 | 7 ] key [
timeout seconds ] | key-chain <name-of-chain> }
```

Task: Debug HSRP authentication

```
# debug standby errors
```

22.1.8. HSRP and Object tracking

- Reduce HSRP priority if the monitored interface goes down, allowing another HSRP router to become active if it has preemption enabled.
- Cumulative reduction if multiple tracked interfaces are down
- Configurable decrement value (default = 10)
- Can shutdown/change the HSRP group to the Init state on the basis of the tracked object's state

Task: Configure interface tracking

```
(config-if)# standby track { <object-number> [<priority-decrement>] | interface-type
<interface-number> [ decrement <priority-decrement> ] } [shutdown]
```

22.1.9. HSRP support for ICMP redirects

- Enabled by default with advertisement every 60 seconds and holddown of 180 seconds

Why?

When HSRP is running, preventing hosts from discovering the interface (or real) IP addresses of devices in the HSRP group is important. If a host is redirected by ICMP to the real IP address of a device, and that device later fails, then packets from the host will be lost.

How?

- looks up the next hop IP address in its table of real IP addresses vs virtual IP address
- if match found, replaces the real IP address by the virtual IP addresses in the gateway field of the redirect packet
- if no match (unknown), send the redirect packet to go out unchanged

Restrictions

- Do not redirect to passive HSRP devices

Task: Enable ICMP redirects on an interface

```
(config-if)# standby redirect [timers <advertisement> <holddown>] [unknown]
```

Task: Disable ICMP redirects on an interface

```
(config-if)# standby redirect [timers <advertisement> <holddown>] [unknown]
```

Task: configure ICMP redirect messages with HSRP virtual IP address as the gateway IP address

```
(config)# standby redirects [enable | disable]
```

Task: Debug HSRP support for ICMP redirects

```
# debug standby events icmp

10:43:08: HSRP: ICMP redirect not sent to 10.0.0.4 for dest 10.0.1.2
10:43:08: HSRP: could not uniquely determine IP address for mac 00d0.bbd3.bc22
```

22.1.10. HSRP virtual IP address and group

- Can have a name (no longer than 25 chars)

Task: Configure the virtual IP address

```
(config-if)# standby [<group-number>] ip [ <a.b.c.d>  [secondary]]
```

- By default, send one gratuitous ARP when a group becomes active and then another two and four seconds later.
- When HSRP is on the Active state on an interface, Proxy ARP requests are answered with the MAC address of the HSRP group. otherwise, they are ignored.

Task: Configure the number of gratuitous ARP packets sent by HSRP group when it transitions to the active state, and how often the ARP packets are sent

```
(config-if)# standby arp gratuitous [count <number=> interval <seconds>]
```

Task: configure the name of the standby group

```
(config-if)# standby name <group-name>
```

22.1.11. Multiple HSRP

- Provides load sharing in an HSRP configuration
  - Because HSRP uses only one Active router at a time, any other HSRP routers are idle.

    - two or more HSRP groups are configured on each HSRP LAN interface, where the configured priority determines which router will be active for each HSRP group.

- requires that each DHCP client and statically configured host are issued a default gateway corresponding to one of the HSRP groups
- requires that they're distributed appropriately.

22.2. GLBP

Configuration Guides | First Hop Redundancy Protocols | Gateway Load Balancing Protocol

- One AVG (active virtual gateway) per group
- Up to 4 primary AVFs (active virtual forwarder)
- Up to 1024 GLBP groups per physical interface
- AVG responds to all ARP requests with MAC from all particapating AVF
    - Not with own MAC address like HSRP
- Load-balancing
    - Round-robin (default)
    - Host-dependent
    - Weigthed
- Multicast, 224.0.0.102, UDP port 3222
- Virtual MAC address 0007.B400.xxyy, where xx the group and yy in (00-03)

22.2.1. GLBP Packet Type

Hello

advertise protocol info, multicast, sent by any AVG or AVF in [Speak, Standby, Active] State

Request

request virtual MAC, unicast sent to AVG

Reply

get virtual MAC, unicast sent from AVG

22.2.2. Active Virtual Gateway

- Elected for each group max(priority, IP address)
- Assigns up to 4 virtual MAC address to AVFs
- Reponds to ARP request for the virtual IP address of the group
- If failed, new AVG election from AVG in listen State
- By default, no AVG preemption

- Load balancing methods
  - Round-robin: by default, each AVF in turn is used for ARP
  - Host-dependent: use same AVF based on the host MAC
  - Weighted: use the weight value advertised by the gateway

Task: Configure GLBP virtual ip address

```
(config-if)# glbp <group:0..1023 > ip [<virtual-ip-address> [secondary]]
```

When the glbp ip command is enabled on an interface, the handling of proxy ARP requests is changed (unless proxy ARP was disabled). the AVG intercepts the ARP requests and replies to the ARP on behalf of the connected nodes. If a forwarder in the GLBP group is active, proxy ARP requests are answered using the MAC address of the first active forwarder in the group. If no forwarder is active, proxy ARP responses are suppressed.

Task: Configure GLBP priority

```
(config-if)# glbp [<group>] priority <level=100>
```

Task: Configure GLBP preemption

```
(config-if)# glbp [<group>] preempt [delay minimum <seconds>]
```

Task: Configure GLBP load balancing

```
(config-if)# glbp <group> load-balancing [ <host-dependent> | <round-robin> |
<weighted> ]
```

Task: Configure the time during which the AVG continues to redirect clients to a secondary AVF

```
(config-if)# glbp <group> timers redirect <redirect-seconds=60> <timeout-seconds=1440>
```

- The redirect timer sets the time delay between a forwarder failing on the network and the AVG assuming that the forwarder will not return. The virtual MAC address to which the forwarder was responsible for replying is still given out in Address Resolution Protocol (ARP) replies, but the forwarding task is handled by another router in the GLBP group.
- The timeout interval is the time delay between a forwarder failing on the network and the MAC address for which the forwarder was responsible becoming inactive on all of the routers in the GLBP group. After the timeout interval, packets sent to this virtual MAC address will be lost. The timeout interval must be long enough to allow all hosts to refresh their ARP cache entry that contained the virtual MAC address.

Task: Display glbp status

```
# sh glbp

FastEthernet0/0.40 - Group 1
  State is Standby
    1 state change, last state change 00:02:02
  Virtual IP address is 172.16.26.100
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.720 secs
  Redirect time 600 sec, forwarder time-out 14400 sec
  Preemption disabled
  Active is 172.16.26.6, priority 100 (expires in 11.360 sec)
  Standby is local
  Priority 100 (default)
  Weighting 100 (default 100), thresholds: lower 1, upper 100
  Load balancing: round-robin
  Group members:
    ca02.6150.0000 (172.16.26.2) local
    ca06.618c.0000 (172.16.26.6)
  There are 2 forwarders (1 active)
  Forwarder 1
    State is Listen
    MAC address is 0007.b400.0101 (learnt)
    Owner ID is ca06.618c.0000
    Time to live: 14399.840 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
    Active is 172.16.26.6 (primary), weighting 100 (expires in 10.944 sec)
  Forwarder 2
    State is Active
      1 state change, last state change 00:02:08
    MAC address is 0007.b400.0102 (default)
    Owner ID is ca02.6150.0000
    Preemption enabled, min delay 30 sec
    Active is local, weighting 100
```

22.2.3. Active Virtual Forwarder

- Primary AVF gets virtual MACs from AVG
- Secondary AVF learns virtual MACs from hellos
- Virtual forwarder preemptive is enabled by default with 30 seconds delay
- Uses weighting and object tracking to determine the forwarding capacity of each device in the GLBP group
  - Decrement or increments the weight when the interface goes down or up
  - Stops being AVF if value below lower threshold
  - Resumes being AVF if value greater then upper threshold
  - When multiple tracked interfaces are down, the configured weighting decrements are cumulative.

Task: Specify GLBP initial weighting value

```
(config-if)# glbp <group-number> weighting <maximum> [lower <low-value> ] [upper <up-value>]
```

Task: specify a tracking object where GLBP weighting changes based on the availability of the object being tracked

```
(config-if)# glbp <group> weighting track <object-number> [decrement <value>]
```

Task: configure a router to take over as (AVF) group if the current AVF falls below its low weighting threshold

```
(config-if)# glbp <group> preempt forwarder [delay minimum <seconds>]
```

22.2.4. Authentication

- Supports no authentication, plain-text, or MD5 authentication

Task: configure glbp authentication

```
(config-if)# glbp authentication { text <string> | key-chain <name> }
```

22.3. VRRP

- **Configuration Guides** | **First Hop Redundancy Protocols** | **VRRP**
- RFC 3768
- Similar to HSRP
- Multicast virtual MAC address (0000.5E00.01xx, where xx is the hex VRRP group number).

- Uses the IOS object tracking feature, rather than its own internal tracking mechanism, to track interface states for failover purposes.
- Preemption by default
- The group IP address is the interface IP address of one of the VRRP routers.

Task: enable VRRP

```
(config-if)# vrrp <group> []
```

Task: verify VRRP configuration

```
# sh vrrp
```

Task: customize VRRP

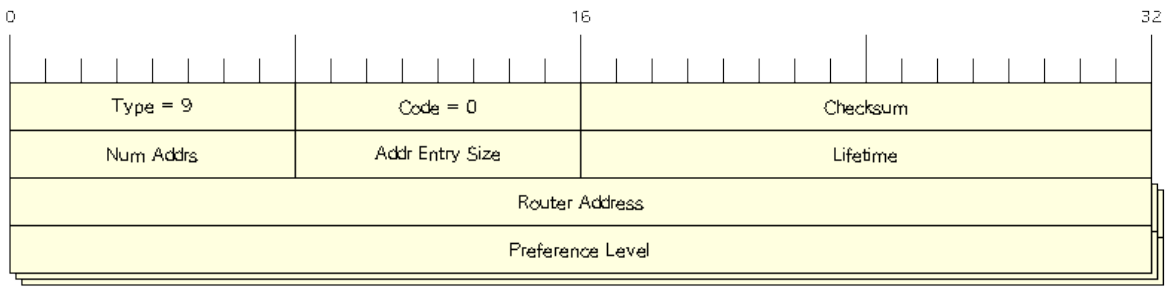
```
conf t
interface <type number>
  ip address ip-address mask
  vrrp <group> description text
  vrrp <group> priority level
  vrrp <group> preempt [delay minimum seconds]
  vrrp <group> timers learn
```

22.4. IDRP

- **Configuration Guides | First Hop Redundancy Protocols | IRDP**
- RFC 1256
- ICMP Router Discovery Protocol allows hosts to locate routers that can be used as gateway to reach IP-based devices on other networks.

22.4.1. Message format

ICMP Router Advertisement Message



Checksum

The 16-bit one's complement of the one's complement sum of the ICMP message, starting with

the ICMP Type. For computing the checksum, the Checksum field is set to 0.

Num Addrs

Number of router addreses advertised in this message

Addr Entry Size

Number of 32-bit words of information per router address (=2 for IPv4)

Lifetime

Maximum number of seconds that the router adresses may be considered valid.

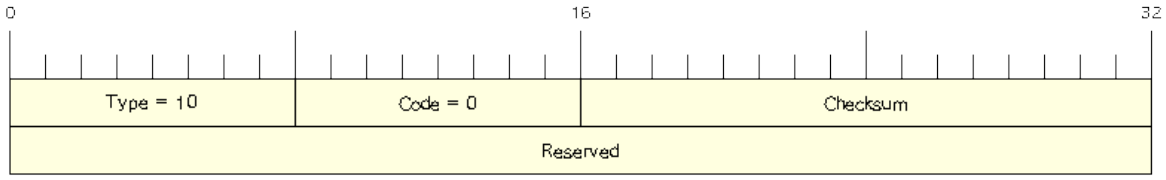
Router Address[i]

Sending router's addresses on the interface from which this message is sent.

Preference Level[i]

Preferability of each Router Address[i] as a default router, relative to other router addresses on the same subnet. Higher values more preferable.

ICMP Router Solicitation Message



Checksum

The 16-bit one's complement of the one's complement sum of the ICMP message, starting with the ICMP Type. For computing the checksum, the Checksum field is set to 0.

Reserved

Sent as 0; ignored on reception.

22.4.2. Configuration

Task: Configure a host to discover routers that transmit IRDP router updates after disabling IP routing

```
(config)# no ip routing
(config-if)# ip gdp irdp [multicast]
```

Task: Enable IRDP on an interface

```
(config-if)# ip irdp
```

Task: Send IRDP Advertisement to the all-systems multicast addresses

```
(config-if)# ip irdp multicast
```

Task: Set the IRDP period for which advertisements are valid.

```
(config-if)# ip irdp holdtime <seconds>
```

Task: Sets the IRDP maximum interval between advertisements.

```
ip irdp maxadvertinterval <seconds>
```

Task: Set the IRDP minimum interval between advertisements.

```
ip irdp minadvertinterval <seconds>
```

Task: Set the IRDP preference level of the device

```
(config-if)# ip irdp preference <number>
```

Task: Specify an IRDP address and preference to proxy-advertise

```
(config-if)# ip irdp address <a.b.c.d> <preference-level>
```

22.5. IPv6 RA/RS

Chapter 23. Multicast

23.1. IGMP

Configuration guides | **Multicast** | **IGMP**

Check also Catalyst configuration guides

23.1.1. Concepts

- Group membership protocol used by hosts to inform routers and multilayer switches of the existence of members on their directly connected networks and to allow them to send and receive multicast datagrams.
- IP protocol number: 2

23.1.2. Messages

- membership queries:
  - general: sent to 224.0.0.1 (all systems on a subnet)
  - group-specific: sent to the group
- membership reports
  - sollicited: sent to the group in v2, sent to 224.0.0.22 in v3
  - unsollicited
- Leave Group messages

23.1.3. Default IGMP configuration

Feature	Default Setting
IGMP version	Version 2 on all interfaces.
IGMP query timeout	60 seconds on all interfaces.
IGMP maximum query response time	10 seconds on all interfaces.
Multilayer switch as a member of a multicast group	No group memberships are defined.
Access to multicast groups	All groups are allowed on an interface.
IGMP host-query message interval	60 seconds on all interfaces.
Multilayer switch as a statically connected member	Disabled.

Task: Display multicast-related information about an interface.

```
# show ip igmp interface [interface-id]
```

IGMP version

v1

- general membership queries
- join
- implicit leave

v2

- group-specific queries
- explicit leave group process
- explicit max response time field
- querier election'

v3

- source filtering
- uses 224.0.0.22 for membership reports

Task: Specify the IGMP version

```
(config-if)# ip igmp version {1 | 2 | 3}
```

Task: Return to the default version

```
(config-if)# no ip igmp version
```

If you change to version 1, you cannot configure the **ip igmp query-interval** or the **ip igmp query-max-response-time** interface configuration commands.

Querier election

- Each IGMPv2 router sends general query message to 224.0.0.1 with its interface source address.
- The router stops upon receiption of query messages with lowest IP address → The router with the lowest IP address wins

IGMPv2 query timeout

- period of time before the multilayer switch takes over as the querier for the interface.
- By default, the switch waits twice the query interval. After that time, if the switch has received no queries, it becomes the querier.

```
(config-if)# ip igmp querier-timeout <60-300-seconds>
```

Maximum response time field

- v1, fixed at 10 seconds
- v2, can be changed to control the burstiness of the response process especially with large number of active routers. Note that increasing the maximum response timer value also increases the leave latency; the query router must now wait longer to make sure there are no more hosts for the group on the subnet.
- Default:10 seconds, range: 1..25.

Task:Change the maximum response time field

```
(config-if)# ip igmp query-max-response-time <seconds>
```

23.1.4. Join the club

Task: Join a specified group

```
(config-if)# ip igmp join-group <address>
```

Task: Join a specified (S,G) channel

```
(config-if)# ip igmp join-group <address> source <a.b.c.d>
```

Task: Display the multicast groups that are directly connected to the multilayer switch and that were learned through IGMP.

```
# sh ip igmp groups [group-name | group-address | type number]
```

Task: Forward multicast packet without accepting them

```
(config-if)# ip igmp static-group
```

This method allows fast switching. The outgoing interface appears in the IGMP cache, but the switch itself is not a member, as evidenced by lack of an L (local) flag in the multicast route entry.

```
(config-if)# ip igmp static-group
```

23.1.5. Leave process

- in v1, implicit exit
- in v2,
  - host send leave group message to group address,

- querier send **igmp-last-member-query-count** group-specific queries at **igmp-last-member-interval** milliseconds
- querier stops forwarding for the group if no reply within timeout period

Task: Specify the last member query interval

```
(config-if)# ip igmp last-member-query-interval <milliseconds>
```

Task: Specify the last member query count

```
(config-if)# ip igmp last-member-query-count <1-7>
```

Task: Minimize the leave latency when only one IGMPv2 receiver is connected to the interface

```
(config-if)# ip igmp immediate-leave group-list <acl>
```

Can also be in global mode but not combined with the interface mode

23.1.6. IGMP message restriction

Task: Restrict receivers on a subnet to join only certain multicast groups

```
(config-if)# ip igmp access-group <standard-acl>
```

Task: Restrict receivers on a subnet to join multicast groups from specific sources

```
(config-if)# ip igmp access-group <extended-acl>
```

23.1.7. IGMP proxy

TODO

23.1.8. IGMP snooping

- Problem: L2 switch forwards multicast packets to all interfaces → wasted traffic
- Solution: Tracks IGMP messages (Join/Leave) to only forward invites to interested parties.
  - Add ports when receiving Join message
  - Delete ports when Leave messages or no membership reports from clients

Table 13. Default IGMP snooping configuration

Feature	Default Setting
IGMP snooping	Enabled globally and per VLAN

Feature	Default Setting
Multicast routers	None configured
Multicast router learning method	PIM-DVMRP
IGMP snooping Immediate Leave	Disabled
Static groups	None configured
TCN flood query count	2
TCN query solicitation	Disabled
IGMP snooping querier	Disabled
IGMP report suppression	Enabled

Task: Display IGMP snooping information

```
# sh ip igmp snooping
```

Task: Disable IGMP snooping globally

```
(config)# no ip igmp snooping
```

Task: Enable VLAN snooping

```
(config)# ip igmp snooping vlan <1-1001,1006-4094>
```

Task: Change the snooping method

```
(config)# ip igmp snooping vlan <vlan-id> mrouter learn {cgmp | pim-dvmrp}
```

Multicast router port

Task: Add a multicast router port

```
(config)# ip igmp snooping vlan <id> mrouter interface <type-number>
```

Task: Verify that IGMP snooping is enabled on the VLAN interface

```
(config)# sh ip igmp snooping mrouter vlan <id>
```

Statically join a group

Task: Add a L2 port to join a group

```
ip igmp snooping vlan <vlan-id> static <ip-address> interface <type number>
```

Hosts or L2 ports normally join multicast groups dynamically

Task: Verify the member port and the IP address

```
# sh ip igmp snooping groups
```

IGMP immediate leave

Task: Remove a port immediately when it detects an IGMPv2 leave message

```
(config)# ip igmp snooping vlan <id> immediate-leave
```

IGMP leave Timer

Task: configure the IGMP leave timer globally

```
ip igmp snooping last-member-query-interval <milliseconds>
```

Task: configure the IGMP leave timer on the VLAN interface

```
ip igmp snooping vlan <id> last-member-query-interval <milliseconds>
```

TCN Events

- when the client changed its loaction and the receiver is on smae port that was blocked but is now forwarding,
- when a port went down without sending a leave message.

Task: Control the multicast flooding time after a TCN event

```
(config)# ip igmp snooping tcn flood query count <1-2-10>
```

Task: Speed the process of recovering from the flood mode caused by a TCN event.

```
(config)# ip igmp snooping tcn query solicit
```

- When a topology change occurs, the spanning-tree root sends a IGMP global leave with group 0.0.0.0.
- however, after **ip igmp snooping tcn query solicit** command, the switch sneds the global leave message whether or not it is the spanning-tree root.
- When the router receives this special leave, it immediately sends general queries, which expedite the process of recovering from the flood mode during the TCN event.

Task: Disable the flooding of multicast traffic during a TCN event

```
(config-if)# no ip igmp snooping tcn flood
```

- When the swith receives a TCN, multicast traffic is flooded to all the ports until 2 general queries are received.
- If the switch has many ports with attached hosts subscribed to many groups, this flooding might exceed the capacity of the link and cause packet loss.

IGMP snooping querier

Task: Enable IGMP snooping querier

```
(config)# ip igmp snooping querier
(config)# ip igmp snooping querier address <ip.ad.re.ss>
(config)# ip igmp snooping querier query-interval <seconds>
(config)# ip igmp snooping querier tcn query [count <n> | interval <seconds>]
(config)# ip igmp snooping querier timer expiry <seconds>
(config)# ip igmp snooping querier version {1 | 2}
```

Task: Display information about the IP address and receiving port for the most-recently received IGMP query in the VLAN

```
# sh ip igmp snooping querier [vlan <id>] [detail]
```

IGMP report suppression

Task: Disable IGMP report suppression

```
(config)# no ip igmp snooping report-suppression
```

23.1.9. MVR

- Multicast VLAN Registration
- Problem: How to scale multicast traffic accross an Ethernet ring-based SP network
- Solution : one multicast VLAN shared with subscribers in seperate VLANs
- Use case: broadcast of multiple TV channels over a service-provider network
- works with or without IGMP snooping
  - If both enabled, MVR reacts only to join and leave messages from MVR groups.

Table 14. Default MVR configuration

Feature	Default Setting
MVR	Disabled globally and per interface
Multicast addresses	None configured
Query response time	0.5 second
Multicast VLAN	VLAN 1
Mode	Compatible
Interface (per port) default	Neither a receiver nor a source port
Immediate Leave	Disabled on all ports

MVR global parameters

Task: Enable MVR on the switch

```
(config)# mvr
```

Task: Configure a range of IP multicast address on the switch

```
(config)# mvr group <ip-address> [count]
```

- The **count** parameter configure a contiguous series of MVR group addresses. Default= 1 in 1..256
- Any multicast data sent to the ip address corresponding to one TV channel is sent to all source ports on the switch and all interested receiver ports.

Task: Define the maximum time to wait for IGMP report memberships on a receiver port

```
(config)# mvr querytime <tenths-of-seconds>
```

Task: Specify the VLAN in which multicast data is received

```
(config)# mvr vlan <vlan-id>
```

- All source ports must belong to this VLAN

Task: Specify the MVR mode of operation

```
(config)# mvr mode { dynamic | compatible }
```

- **dynamic**: allows dynamic MVR memberships on source ports.
- **compatible** is the default and does not support ICMP dynamic joins on source ports.

Task: Verify the MVR global configuration

```
(config)# sh mvr
(config)# sh mvr members
```

MVR interfaces

Task: configure an MVR port as source

```
(config-if)# mvr type source
```

- Configure uplinks ports that receive and send multicast data as source ports
- Subscribers cannot be directly connected to source ports
- All source ports on a switc belong to the single multicast VLAN.

Task: configure an MVR port as receiver

```
(config-if)# mvr type receiver
```

- Configure a port as a receiver port if it is a subscriber port and should only recieve multicast data.
- Receiver ports do not receive data unless it becomes a member of the multicast group.
- Receiver ports cannot belongs to the multicast VLAN.

Task: Statically configure a port to receive multicast traffic

```
(config)# mvr vlan <id> group <ip-address>
```

Task: Enable the Immediate-Leave feature of MVR on the receiver port

```
(config)# mvr immediate
```

Task: Verify the MVR interface configuration

```
# sh mvr interface
```

Task: Display all receiver and source ports that are members of a multicast group

```
# sh mvr members [group-ip-address]
```

23.1.10. IGMP filtering and throttling

Table 15. Default IGMP filtering configuration

Feature	Default Setting
Filters	none applied
profiles	none defined
profile action	deny the range addresses

IGMP profiles

Task: Configure an IGMP profile

```
(config)# ip igmp profile <number>
(config-igmp-profile)# permit | deny
(config-igmp-profile)# range <low-ip-address> [<high-ip-address>]
```

Task: Apply IGMP profile to an interface

```
(config)# ip igmp filter <profile-number>
```

Task: Verify the profile configuration

```
# sh ip igmp profile <number>
```

IGMP throttling

Task: Set the maximum number of IGMP groups that the interface can join

```
(config-if)# ip igmp max-groups <count>
```

Task: Specify the action that the interface takes when it reaches the maximum number of entries and receives a new IGMP report

```
(config-if)# ip igmp max-groups action {deny | replace }
```

23.2. PIM

23.2.1. Understanding

- protocol-independent
  - relies on unicast routing table to perform RPF check

Versions

PIMv1

- Introduced in IOS 11.1(6)
- Support Auto-RP : eliminates the need to manually configure the rendezvous point in every router.

PIMv2

- Default version since IOS 11.3
- Support BSR (boot strap router) capability.
- A single, active RP exists per multicast group, with multiple backup RPs. This single RP compares to multiple active RPs for the same group in PIMv1.
- A BSR provides a fault-tolerant, automated RP discovery and distribution mechanism that enables routers and multilayer switches to dynamically learn the group-to-RP mappings.
- Sparse mode and dense mode are properties of a group, as opposed to an interface. We strongly recommend sparse-dense mode, as opposed to either sparse mode or dense mode only.
- PIM join and prune messages have more flexible encoding for multiple address families.
- A more flexible hello packet format replaces the query packet to encode current and future capability options.
- Register messages to an RP specify whether they are sent by a border router or a designated router.
- PIM packets are no longer inside IGMP packets; they are standalone packets.

Modes

PIM can operate in dense mode (DM), sparse mode (SM), or in sparse-dense mode (PIM DM-SM), which handles both sparse groups and dense groups at the same time.

PIM DM

In dense mode, a PIM DM router assumes that all other routers es forward multicast packets for a group. If a PIM DM device receives a multicast packet and has no directly connected members or PIM neighbors present, a prune message is sent back to the source. Subsequent multicast packets are not flooded to this router or switch on this pruned branch. PIM DM builds source-based multicast distribution trees.

The simplest form of a multicast distribution tree is a source tree whose root is the source of the multicast traffic and whose branches form a spanning tree through the network to the receivers. Because this tree uses the shortest path through the network, it is also referred to as a shortest-path

tree (SPT). A separate SPT exists for every individual source sending to each group. The special notation of (S,G) (pronounced S comma G) identifies an SPT where S is the IP address of the source and G is the multicast group address.

Host A Shortest-Path Tree shows an example of SPT for group 224.1.1.1 rooted at the source, Host A, and connecting two receivers, Hosts B and C. The SPT notation for this group would be (194.1.1.1, 224.1.1.1).

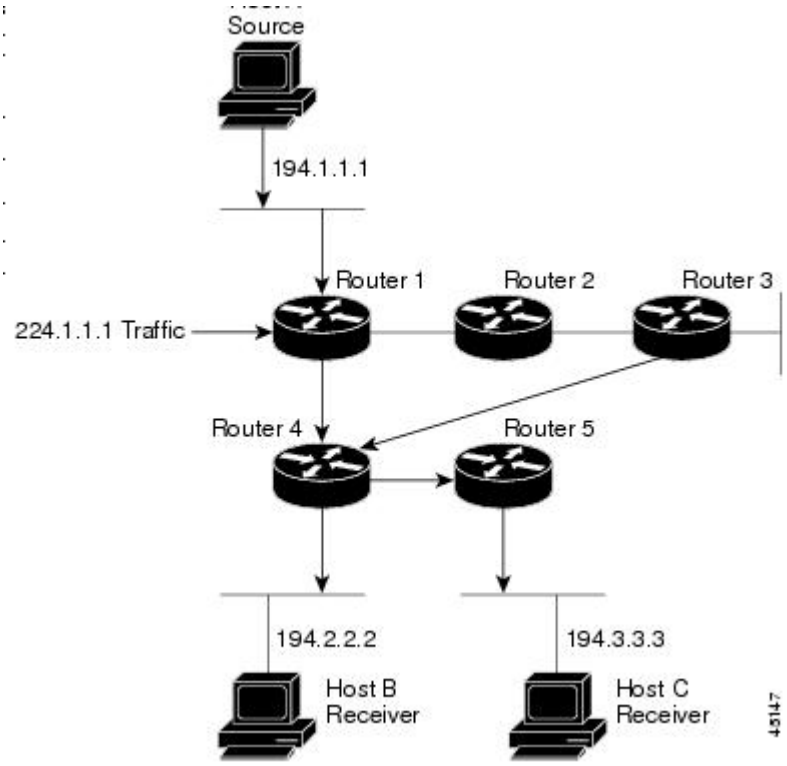


Figure 33. Host A Shortest-Path Tree

If Host B is also sending traffic to group 224.1.1.1 and Hosts A and C are receivers, then a separate (S,G) SPT would exist with the notation of (194.2.2.2, 224.1.1.1).

PIM DM employs only SPTs to deliver (S,G) multicast traffic by using a flood and prune method. It assumes that every subnet in the network has at least one receiver of the (S,G) multicast traffic, and therefore the traffic is flooded to all points in the network.

To avoid unnecessary consumption of network resources, PIM DM devices send prune messages up the source distribution tree to stop unwanted multicast traffic. Branches without receivers are pruned from the distribution tree, leaving only branches that contain receivers. Prunes have a timeout value associated with them, after which the PIM DM device puts the interface into the forwarding state and floods multicast traffic out the interface. When a new receiver on a previously pruned branch of the tree joins a multicast group, the PIM DM device detects the new receiver and immediately sends a graft message up the distribution tree toward the source. When the upstream PIM DM device receives the graft message, it immediately puts the interface on which the graft was received into the forwarding state so that the multicast traffic begins flowing to the receiver.

PIM SM

PIM SM uses shared trees and SPTs to distribute multicast traffic to multicast receivers in the network.

In PIM SM, a router assumes that other routers or switches do not forward multicast packets for a group, unless there is an explicit request for the traffic (join message). When a host joins a multicast group using IGMP, its directly connected PIM SM device sends PIM join messages toward the root, also known as the RP. This join message travels router-by-router toward the root, constructing a branch of the shared tree as it goes. The RP keeps track of multicast receivers; it also registers sources through register messages received from the source's first-hop router (designated router [DR]) to complete the shared tree path from the source to the receiver. The branches of the shared tree are maintained by periodic join refresh messages that the PIM SM devices send along the branch.

When using a shared tree, sources must send their traffic to the RP so that the traffic reaches all receivers. The special notation *,G, (pronounced star comma G) is used to represent the tree, where * means all sources and G represents the multicast group. Figure Shared Distribution Tree shows a shared tree for group 224.2.2.2 with the RP located at Router 3. Multicast group traffic from source Hosts A and D travels to the RP (Router 3) and then down the shared tree to two receivers, Hosts B and C. Because all sources in the multicast group use a common shared tree, the special notation (*, 224.2.2.2) describes this shared tree.

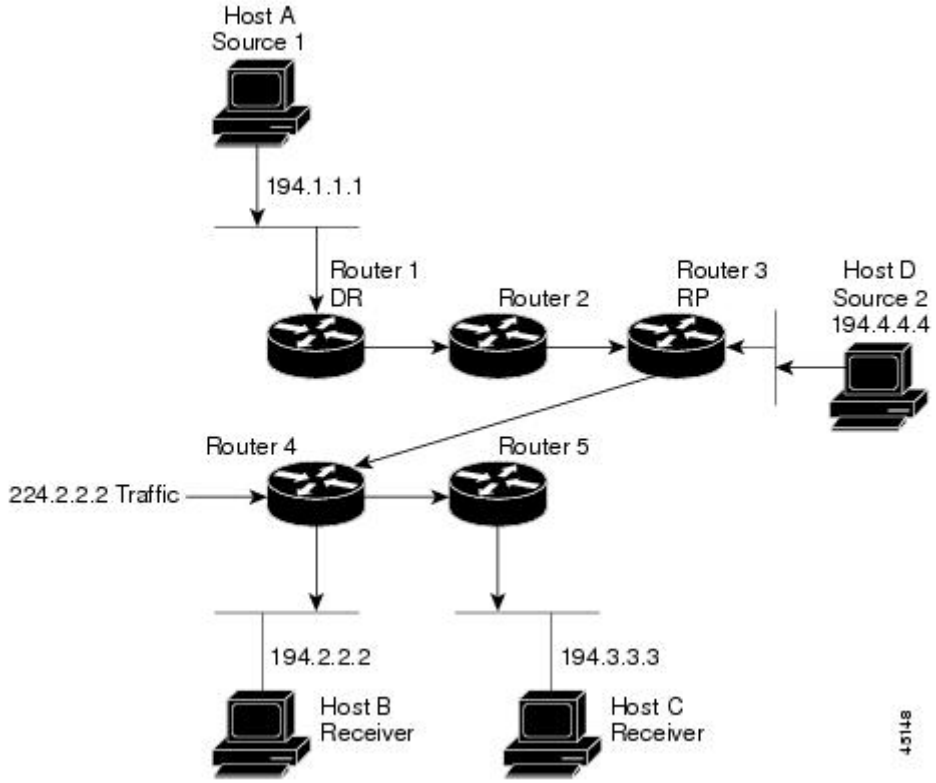


Figure 34. Shared Distribution Tree

In addition to using the shared distribution tree, PIM SM can also use SPTs. By joining an SPT, multicast traffic is routed directly to the receivers without having to go through the RP, thereby reducing network latency and possible congestion at the RP. The disadvantage is that PIM SM devices must create and maintain (S,G) state entries in their routing tables along with the (S,G) SPT. This action consumes router resources.

Prune messages are sent up the distribution tree to prune multicast group traffic. This action permits branches of the shared tree or SPT that were created with explicit join messages to be torn

down when they are no longer needed. For example, if a leaf router (a router without any downstream connections) detects that it no longer has any directly connected hosts (or downstream multicast routers) for a particular multicast group, it sends a prune message up the distribution tree to stop the flow of unwanted multicast traffic.

Shared tree vs Source tree

By default, members of a group receive data from senders to the group across a single data-distribution tree rooted at the RP. Figure PIM trees shows this type of shared-distribution tree. Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

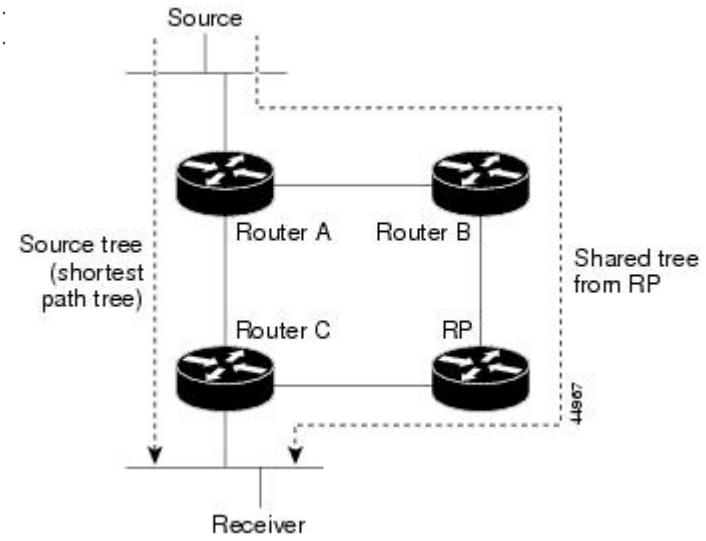


Figure 35. PIM trees

If the data rate warrants, leaf routers (routers without any downstream connections) on the shared tree can use the data distribution tree rooted at the source. This type of distribution tree is called a shortest-path tree or source tree. By default, the IOS software switches to a source tree upon receiving the first data packet from a source.

This process describes the move from a shared tree to a source tree:

1. A receiver joins a group; leaf Router C sends a join message toward the RP.
2. The RP puts a link to Router C in its outgoing interface list.
3. A source sends data; Router A encapsulates the data in a register message and sends it to the RP.
4. The RP forwards the data down the shared tree to Router C and sends a join message toward the source. At this point, data might arrive twice at Router C, once encapsulated and once natively.
5. When data arrives natively (unencapsulated) at the RP, it sends a register-stop message to Router A.
6. By default, reception of the first data packet prompts Router C to send a join message toward the source.
7. When Router C receives data on (S,G), it sends a prune message for the source up the shared tree.
8. The RP deletes the link to Router C from the outgoing interface of (S,G). The RP triggers a prune message toward the source.

Join and prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM device along the path to the source or RP. Register and register-stop messages are not sent hop-by-hop. They are sent by the designated router that is directly connected to a source and are received by the RP for the group.

Multiple sources sending to groups use the shared tree.

You can configure the PIM device to stay on the shared tree.

Auto-RP

This proprietary feature eliminates the need to manually configure the rendezvous point (RP) information in every router and multilayer switch in the network. Auto-RP uses IP multicast to automate the distribution of group-to-RP mappings to all Cisco routers and multilayer switches in a PIM network.

It has these benefits:

- It is easy to use multiple RPs within a network to serve different group ranges.
- It allows load splitting among different RPs and arrangement of RPs according to the location of group participants.
- It avoids inconsistent, manual RP configurations on every router and multilayer switch in a PIM network, which can cause connectivity problems.

For Auto-RP to work, you configure a Cisco router as the **mapping agent**. It uses IP multicast to learn which routers or switches in the network are possible candidate RPs by joining the well-known Cisco-RP-announce multicast group (224.0.1.39) to receive candidate RP announcements. Candidate RPs send multicast RP-announce messages to a particular group or group range every 60 seconds (default) to announce their availability. Each RP-announce message contains a holdtime that tells the mapping agent how long the candidate RP announcement is valid. The default is 180 seconds.

Mapping agents listen to these candidate RP announcements and use the information to create entries in their Group-to-RP mapping caches. Only one mapping cache entry is created for any Group-to-RP range received, even if multiple candidate RPs are sending RP announcements for the same range. As the RP-announce messages arrive, the mapping agent selects the router or switch with the highest IP address as the active RP and stores this RP address in the Group-to-RP mapping cache.

Mapping agents multicast the contents of their Group-to-RP mapping cache in RP-discovery messages every 60 seconds (default) to the Cisco-RP-discovery multicast group (224.0.1.40), which all Cisco PIM routers and multilayer switches join to receive Group-to-RP mapping information. Thus, all routers and switches automatically discover which RP to use for the groups they support. The discovery messages also contain a holdtime, which defines how long the Group-to-RP mapping is valid. If a router or switch fails to receive RP-discovery messages and the Group-to-RP mapping information expires, it switches to a statically configured RP that was defined with the **ip pim rp-address** global configuration command. If no statically configured RP exists, the router or switch changes the group to dense-mode operation.

Multiple RPs serve different group ranges or serve as hot backups of each other.

Bootstrap Router

PIMv2 BSR is another method to distribute group-to-RP mapping information to all PIM routers and multilayer switches in the network. It eliminates the need to manually configure RP information in every router and switch in the network. However, instead of using IP multicast to distribute group-to-RP mapping information, BSR uses hop-by-hop flooding of special BSR messages to distribute the mapping information.

The BSR is elected from a set of candidate routers and switches in the domain that have been configured to function as BSRs. The election mechanism is similar to the root-bridge election mechanism used in bridged LANs. The BSR election is based on the BSR priority of the device contained in the BSR messages that are sent hop-by-hop through the network. Each BSR device examines the message and forwards out all interfaces only the message that has either a higher BSR priority than its BSR priority or the same BSR priority, but with a higher BSR IP address. Using this method, the BSR is elected.

The elected BSR sends BSR messages to the all-PIM-routers multicast group (224.0.0.13) with a TTL of 1. Neighboring PIMv2 routers es receive the BSR message and multicast it out all other interfaces (except the one on which it was received) with a TTL of 1. In this way, BSR messages travel hop-by-hop throughout the PIM domain. Because BSR messages contain the IP address of the current BSR, the flooding mechanism allows candidate RPs to automatically learn which device is the elected BSR.

Candidate RPs send candidate RP advertisements showing the group range for which they are responsible directly to the BSR, which stores this information in its local candidate-RP cache. The BSR periodically advertises the contents of this cache in BSR messages to all other PIM devices in the domain. These messages travel hop-by-hop through the network to all routers and switches, which store the RP information in the BSR message in their local RP cache. The routers and switches select the same RP for a given group because they all use a common RP hashing algorithm.

Multicast Forwarding and Reverse Path Check

With unicast routing, routers and multilayer switches forward traffic through the network along a single path from the source to the destination host whose IP address appears in the destination address field of the IP packet. Each router and switch along the way makes a unicast forwarding decision, using the destination IP address in the packet, by looking up the destination address in the unicast routing table and forwarding the packet through the specified interface to the next hop toward the destination.

With multicasting, the source is sending traffic to an arbitrary group of hosts represented by a multicast group address in the destination address field of the IP packet. To determine whether to forward or drop an incoming multicast packet, the router uses a **reverse path forwarding** (RPF) check on the packet as follows and shown in Figure RPF Check:

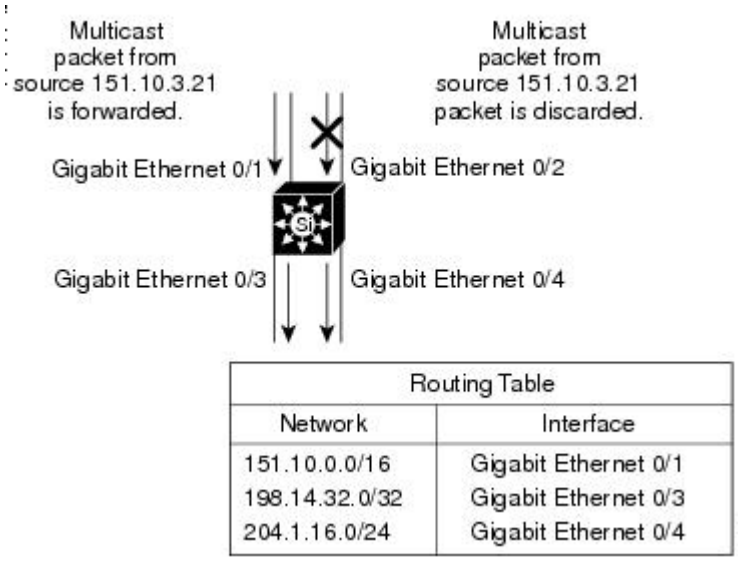
1. The router examines the source address of the arriving multicast packet to determine whether the packet arrived on an interface that is on the reverse path back to the source.
2. If the packet arrives on the interface leading back to the source, the RPF check is successful and

the packet is forwarded to all interfaces in the outgoing interface list (which might not be all interfaces on the router).

3. If the RPF check fails, the packet is discarded.

Some multicast routing protocols, such as DVMRP, maintain a separate multicast routing table and use it for the RPF check. However, PIM uses the unicast routing table to perform the RPF check.

Figure RPF Check shows Gigabit Ethernet interface 0/2 receiving a multicast packet from source 151.10.3.21. A check of the routing table shows that the interface on the reverse path to the source is Gigabit Ethernet interface 0/1, not interface 0/2. Because the RPF check fails, the multilayer switch discards the packet. Another multicast packet from source 151.10.3.21 is received on interface 0/1, and the routing table shows this interface is on the reverse path to the source. Because the RPF check passes, the switch forwards the packet to all interfaces in the outgoing interface list.



Routing Table	
Network	Interface
151.10.0.0/16	Gigabit Ethernet 0/1
198.14.32.0/32	Gigabit Ethernet 0/3
204.1.16.0/24	Gigabit Ethernet 0/4

Figure 36. RPF Check

PIM uses both source trees and RP-rooted shared trees to forward datagrams ; the RPF check is performed differently for each:

- If a PIM router has a source-tree state (that is, an (S,G) entry is present in the multicast routing table), it performs the RPF check against the IP address of the source of the multicast packet.
- If a PIM router has a shared-tree state (and no explicit source-tree state), it performs the RPF check on the rendezvous point (RP) address (which is known when members join the group).

Sparse-mode PIM uses the RPF lookup function to determine where it needs to send joins and prunes:

- (S,G) joins (which are source-tree states) are sent toward the source.
- (*,G) joins (which are shared-tree states) are sent toward the RP.

DVMRP and dense-mode PIM use only source trees and use RPF as previously described.

Neighbor Discovery

PIM uses a neighbor discovery mechanism to establish PIM neighbor adjacencies. To establish

adjacencies, a PIM router sends PIM hello messages to the all-PIM-routers multicast group (224.0.0.13) on each of its multicast-enabled interfaces. The hello message contains a holdtime, which tells the receiver when the neighbor adjacency associated with the sender expires if no more PIM hello messages are received. Keeping track of adjacencies is important for PIM DM operation for building the source distribution tree.

PIM hello messages are also used to elect the DR for multi-access networks (Ethernet). The router on the network with the highest IP address is the DR. With PIM DM operation, the DR has meaning only if IGMPv1 is in use; IGMPv1 does not have an IGMP querier election process, so the elected DR functions as the IGMP querier. In PIM SM operation, the DR is the router or switch that is directly connected to the multicast source. It sends PIM register messages to notify the RP that multicast traffic from a source needs to be forwarded down the shared tree.

PIMv1 and PIMv2 interoperability

The Cisco PIMv2 implementation allows interoperability and transition between Version 1 and Version 2, although there might be some minor problems.

You can upgrade to PIMv2 incrementally. PIM Versions 1 and 2 can be configured on different routers and multilayer switches within one network. Internally, all routers and multilayer switches on a shared media network must run the same PIM version. Therefore, if a PIMv2 device detects a PIMv1 device, the Version 2 device downgrades itself to Version 1 until all Version 1 devices have been shut down or upgraded.

PIMv2 uses the BSR to discover and announce RP-set information for each group prefix to all the routers and multilayer switches in a PIM domain. PIMv1, together with the Auto-RP feature, can perform the same tasks as the PIMv2 BSR. However, Auto-RP is a standalone protocol, separate from PIMv1, and is a proprietary Cisco protocol. PIMv2 is a standards track protocol in the IETF. We recommend that you use PIMv2. The BSR mechanism interoperates with Auto-RP on Cisco routers and multilayer switches.

When PIMv2 devices interoperate with PIMv1 devices, Auto-RP should have already been deployed. A PIMv2 BSR that is also an Auto-RP mapping agent automatically advertises the RP elected by Auto-RP. That is, Auto-RP sets its single RP on every router in the group. Not all routers and switches in the domain use the PIMv2 hash function to select multiple RPs.

Dense-mode groups in a mixed PIMv1 and PIMv2 region need no special configuration; they automatically interoperate.

Sparse-mode groups in a mixed PIMv1 and PIMv2 region are possible because the Auto-RP feature in PIMv1 interoperates with the PIMv2 RP feature. Although all PIMv2 devices can also use PIMv1, we recommend that the RPs be upgraded to PIMv2 (or at least upgraded to PIMv1 in the Cisco IOS Release 11.3 software). To ease the transition to PIMv2, we have these recommendations:

- Use Auto-RP throughout the region.
- Configure sparse-dense mode throughout the region.

Auto-RP and BSR configuration guidelines

There are two approaches to using PIMv2. You can use Version 2 exclusively in your network or migrate to Version 2 by employing a mixed PIM version environment.

- If your network is all Cisco routers and multilayer switches, you can use either Auto-RP or BSR.
- If you have non-Cisco routers in your network, you must use BSR.
- If you have Cisco PIMv1 and PIMv2 routers and multilayer switches and non-Cisco routers, you must use both Auto-RP and BSR.
- Because bootstrap messages are sent hop-by-hop, a PIMv1 device prevents these messages from reaching all routers and multilayer switches in your network. Therefore, if your network has a PIMv1 device in it and only Cisco routers and multilayer switches, it is best to use Auto-RP.
- If you have a network that includes non-Cisco routers, configure the Auto-RP mapping agent and the BSR on a Cisco PIMv2 router . Ensure that no PIMv1 device is on the path between the BSR and a non-Cisco PIMv2 router.
- If you have non-Cisco PIMv2 routers that need to interoperate with Cisco PIMv1 routers and multilayer switches, both Auto-RP and a BSR are required. We recommend that a Cisco PIMv2 device be both the Auto-RP mapping agent and the BSR.

23.2.2. Configuration tasks

Configure basic multicast routing

You must enable IP multicast routing and configure the PIM version and PIM mode so that the IOS software can forward multicast packets and determine how the multilayer switch populates its multicast routing table.

You can configure an interface to be in PIM dense mode, sparse mode, or sparse-dense mode. The mode determines how the switch populates its multicast routing table and how it forwards multicast packets it receives from its directly connected LANs. You must enable PIM in one of these modes for an interface to perform IP multicast routing. Enabling PIM on an interface also enables IGMP operation on that interface.

By default, multicast routing is disabled, and there is no default mode setting. The following procedure is required.

- Enable IP multicast forwarding

```
(config)# ip multicast-routing
```

- Enter interface configuration mode, and specify the Layer 3 interface on which you want to enable multicast routing. The specified interface must be one of the following:
  - A routed port: a physical port that has been configured as a Layer 3 port by entering the no **switchport interface** configuration command.
  - An SVI: a VLAN interface created by using the **interface vlan vlan-id** global configuration command.

These ports must have IP addresses assigned to them.

```
interface <interface-id>
```

- Configure the PIM version on the interface. By default, version 2 is enabled and is the recommended setting.

```
(config-if)# ip pim version [1|2]
```

- Enable a PIM mode on the interface. By default, no mode is configured.

```
(config-if)# pim {dense-mode | sparse-mode | sparse-dense-mode }
```

Manually Assigning an RP to Multicast Groups

Senders of multicast traffic announce their existence through register messages received from the source's first-hop router (designated router) and forwarded to the RP. Receivers of multicast packets use RPs to join a multicast group by using explicit join messages. RPs are not members of the multicast group; rather, they serve as a meeting place for multicast sources and group members.

Configure the address of a PIM RP.

By default, no PIM RP address is configured. You must configure the IP address of RPs on all routers and multilayer switches (including the RP). If there is no RP configured for a group, the multilayer switch treats the group as dense, using the dense-mode PIM techniques. A PIM device can use multiple RPs, but only one per group.

- For ip-address, enter the unicast address of the RP in dotted-decimal notation.
- (Optional) For access-list-number, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups.
- (Optional) The override keyword means that if there is a conflict between the RP configured with this command and one learned by Auto-RP or BSR, the RP configured with this command prevails.

```
ip pim rp-address ip-address [access-list-number] [override]
```

Configure Auto-RP

Configure another PIM device to be the candidate RP for local groups.

- For interface-id, enter the interface type and number that identifies the RP address. Valid interfaces include physical ports, port channels, and VLANs.
- For scope ttl, specify the time-to-live value in hops. Enter a hop count that is high enough so that the RP-announce messages reach all mapping agents in the network. There is no default setting.

The range is 1 to 255.

- For **group-list** access-list-number, enter an IP standard access list number from 1 to 99. If no access list is configured, the RP is used for all groups.
- For interval seconds, specify how often the announcement messages must be sent. The default is 60 seconds. The range is 1 to 16383.

```
ip pim send-rp-announce <interface-id> scope <ttl> group-list <access-list-number>
interval <seconds>
```

Find a multilayer switch whose connectivity is not likely to be interrupted, and assign it the role of RP-mapping agent.

For scope ttl, specify the time-to-live value in hops to limit the RP discovery packets. All devices within the hop count from the source device receive the Auto-RP discovery messages. These messages tell other devices which group-to-RP mapping to use to avoid conflicts (such as overlapping group-to-RP ranges). There is no default setting. The range is 1 to 255.

```
ip pim send-rp-discovery scope <1..255>
```

Configure PIM-SM interfaces to use dense mode to flood Auto-RP traffic to 224.0.1.39 and 224.0.1.40.

```
ip pim autorp listener
```

Prevent Join Messages to false RPs

Determine whether the **ip pim accept-rp** command was previously configured throughout the network by using the show running-config privileged EXEC command. If the **ip pim accept-rp** command is not configured on any device, this problem can be addressed later. In those routers es already configured with the **ip pim accept-rp** command, you must enter the command again to accept the newly advertised RP.

To accept all RPs advertised with Auto-RP and reject all other RPs by default, use the **ip pim accept-rp auto-rp** global configuration command.

If all interfaces are in sparse mode, use a default-configured RP to support the two well-known groups 224.0.1.39 and 224.0.1.40. Auto-RP uses these two well-known groups to collect and distribute RP-mapping information. When this is the case and the **ip pim accept-rp auto-rp** command is configured, another **ip pim accept-rp** command accepting the RP must be configured as follows:

```
Switch(config)# ip pim accept-rp 172.10.20.1 1
Switch(config)# access-list 1 permit 224.0.1.39
Switch(config)# access-list 1 permit 224.0.1.40
```

Prevent candidate RP spoofing

Filter incoming RP announcement messages.

Enter this command on each mapping agent in the network.

Without this command, all incoming RP-announce messages are accepted by default.

For **rp-list** access-list-number, configure an access list of candidate RP addresses that, if permitted, is accepted for the group ranges supplied in the group-list access-list-number variable. If this variable is omitted, the filter applies to all multicast groups.

If more than one mapping agent is used, the filters must be consistent across all mapping agents to ensure that no conflicts occur in the Group-to-RP mapping information.

```
ip pim rp-announce-filter rp-list <access-list-number> group-list <access-list-number>
```

Define the PIM domain border

As IP multicast becomes more widespread, the chances of one PIMv2 domain bordering another PIMv2 domain is increasing. Because these two domains probably do not share the same set of RPs, BSR, candidate RPs, and candidate BSRs, you need to constrain PIMv2 BSR messages from flowing into or out of the domain. Allowing these messages to leak across the domain borders could adversely affect the normal BSR election mechanism and elect a single BSR across all bordering domains and co-mingle candidate RP advertisements, resulting in the election of RPs in the wrong domain.

Define a PIM bootstrap message boundary for the PIM domain.

Enter this command on each interface that connects to other bordering PIM domains. This command instructs the multilayer switch to neither send or receive PIMv2 BSR messages on this interface as shown in Figure Constraining PIMv2 BSR Messages.

```
(config-if)# ip pim bsr-border
```

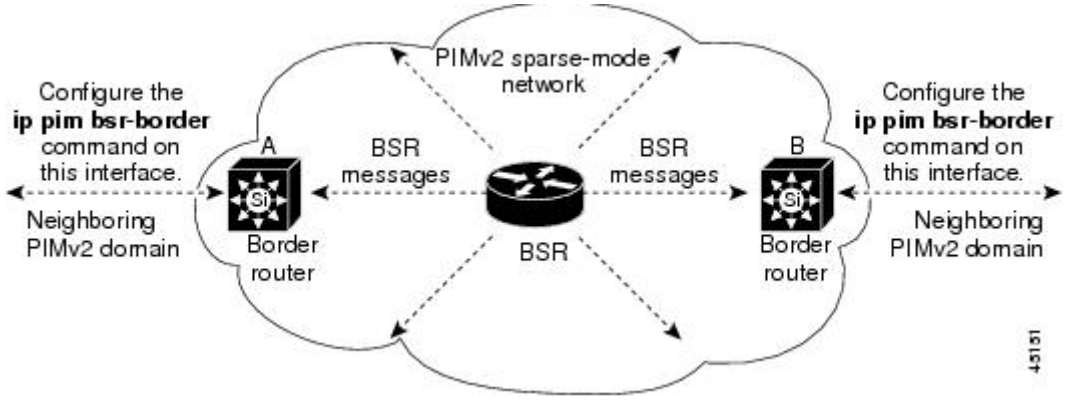


Figure 37. Constraining PIMv2 BSR Messages

Define the IP multicast boundary

You define a multicast boundary to prevent Auto-RP messages from entering the PIM domain. You create an access list to deny packets destined for 224.0.1.39 and 224.0.1.40, which carry Auto-RP information.

```
(config-if)# ip multicast boundary <access-list-number>
```

Configure candidate BSRs

You can configure one or more candidate BSRs. The devices serving as candidate BSRs should have good connectivity to other devices and be in the backbone portion of the network.

Configure your multilayer switch to be a candidate BSR.

- For interface-id, enter the interface type and number on this switch from which the BSR address is derived to make it a candidate. This interface must be enabled with PIM. Valid interfaces include physical ports, port channels, and VLANs.
- For hash-mask-length, specify the mask length (32 bits maximum) that is to be ANDed with the group address before the hash function is called. All groups with the same seed hash correspond to the same RP. For example, if this value is 24, only the first 24 bits of the group addresses matter.
- (Optional) For priority, enter a number from 0 to 255. The BSR with the larger priority is preferred. If the priority values are the same, the device with the highest IP address is selected as the BSR. The default is 0.

```
(config)# ip pim bsr-candidate <interface-id> <hash-mask-length> [priority]
```

Configure Candidate RPs

You can configure one or more candidate RPs. Similar to BSRs, the RPs should also have good connectivity to other devices and be in the backbone portion of the network. An RP can serve the entire IP multicast address space or a portion of it. Candidate RPs send candidate RP advertisements to the BSR. When deciding which devices should be RPs, consider these options:

- In a network of Cisco routers and multilayer switches where only Auto-RP is used, any device can be configured as an RP.
- In a network that includes only Cisco PIMv2 routers and multilayer switches and with routers from other vendors, any device can be used as an RP.
- In a network of Cisco PIMv1 routers, Cisco PIMv2 routers, and routers from other vendors, configure only Cisco PIMv2 routers and multilayer switches as RPs.

Configure your multilayer switch to be a candidate RP.

- For interface-id, enter the interface type and number whose associated IP address is advertised as a candidate RP address. Valid interfaces include physical ports, port channels, and VLANs.

- (Optional) For group-list access-list-number, enter an IP standard access list number from 1 to 99. If no group-list is specified, the multilayer switch is a candidate RP for all groups.

```
ip pim rp-candidate interface-id [group-list access-list-number]
```

Delay the Use of PIM Shortest-Path Tree

The change from shared to source tree happens when the first data packet arrives at the last-hop router. This change occurs because the **ip pim spt-threshold** interface configuration command controls that timing; its default setting is 0 kbps.

The shortest-path tree requires more memory than the shared tree but reduces delay. You might want to postpone its use. Instead of allowing the leaf router to immediately move to the shortest-path tree, you can specify that the traffic must first reach a threshold.

You can configure when a PIM leaf router should join the shortest-path tree for a specified group. If a source sends at a rate greater than or equal to the specified kbps rate, the multilayer switch triggers a PIM join message toward the source to construct a source tree (shortest-path tree). If the traffic rate from the source drops below the threshold value, the leaf router switches back to the shared tree and sends a prune message toward the source.

You can specify to which groups the shortest-path tree threshold applies by using a group list (a standard access list). If a value of 0 is specified or if the group list is not used, the threshold applies to all groups.

Specify the threshold that must be reached before moving to shortest-path tree (spt).

- For kbps, specify the traffic rate in kilobits per second. The default is 0 kbps. The range is 0 to 4294967.
- Specify infinity if you want all sources for the specified group to use the shared tree, never switching to the source tree.
- (Optional) For group-list access-list-number, specify the access list created in Step 2. If the value is 0 or if the group-list is not used, the threshold applies to all groups.

```
ip pim spt-threshold {kbps | infinity} [group-list access-list-number]
```

Modifying the PIM Router-Query Message Interval

PIM routers and multilayer switches send PIM router-query messages to determine which device will be the DR for each LAN segment (subnet). The DR is responsible for sending IGMP host-query messages to all hosts on the directly connected LAN.

With PIM DM operation, the DR has meaning only if IGMPv1 is in use. IGMPv1 does not have an IGMP querier election process, so the elected DR functions as the IGMP querier. With PIM SM operation, the DR is the device that is directly connected to the multicast source. It sends PIM register messages to notify the RP that multicast traffic from a source needs to be forwarded down the shared tree. In this case, the DR is the device with the highest IP address.

The default is 30 seconds. The range is 1 to 65535.

```
ip pim query-interval <seconds>
```

Verify

Display information about interfaces configured for PIM.

```
show ip pim interface [type number] [count]
```

List the PIM neighbors discovered by the multilayer switch.

```
show ip pim neighbor [type number]
```

Display the elected BSR

```
show ip pim bsr
```

displays the RP that was selected for the specified group.

```
show ip pim rp-hash group
```

displays how the multilayer switch learns of the RP (through the BSR or the Auto-RP mechanism).

```
show ip pim rp [group-name | group-address | mapping]
```

Display the RP routers associated with a sparse-mode multicast group.

```
show ip pim rp [group-name | group-address]
```

Display how the multilayer switch is doing Reverse-Path Forwarding

```
show ip rpf {source-address | name}
```

Query a multicast router about which neighboring multicast devices are peering with it.

```
mrinfo [hostname | address] [source-address | interface]
```

Display IP multicast packet rate and loss information.

```
mstat source [destination] [group]
```

Trace the path from a source to a destination branch for a multicast distribution tree for a given group.

```
mtrace source [destination] [group]
```

23.2.3. Troubleshoot

When debugging interoperability problems between PIMv1 and PIMv2, check these in the order shown:

1. Verify RP mapping with the show ip pim rp-hash privileged EXEC command, making sure that all systems agree on the same RP for the same group.
2. Verify interoperability between different versions of DRs and RPs. Make sure the RPs are interacting with the DRs properly (by responding with register-stops and forwarding decapsulated data packets from registers).

Load splitting IP multicast traffic over ECMP

23.2.4. Misc

TODO To be added in the text

Table 16. PIM type code

Type	Name
0	Hello
1	Register
2	Register Stop
3	Join/Prune
4	Bootstrap
5	Assert
6	Graft
7	Graft-Ack
8	Candidate RP Advertisement
9	State Refresh
10	DF Election
11-14	Unassigned
15	Reserved for extension of type space

Define the ssm range of IP multicast addresses

```
(config)# ip pim [vrf name] ssm { default | range access-list-number}
```

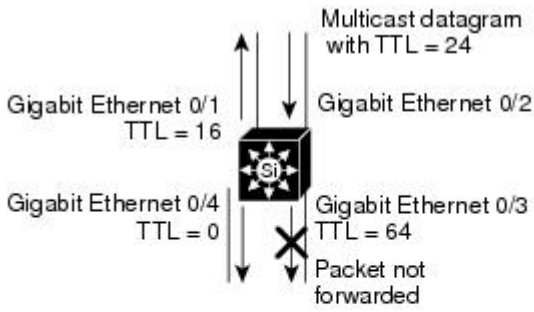
default defines the ssm range access list to 232/8

TODO: Need to delete section below.

Configure the TTL Threshold

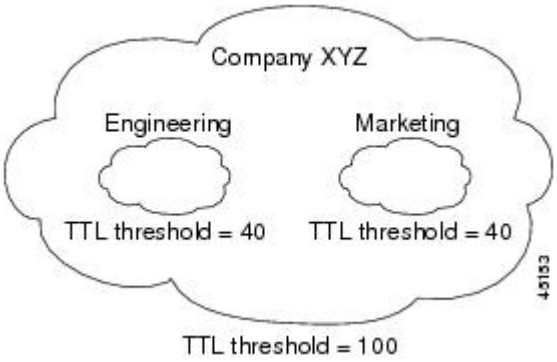
Each time an IP multicast packet is forwarded by the multilayer switch, the time-to-live (TTL) value in the IP header is decremented by one. If the packet TTL decrements to zero, the switch drops the packet. TTL thresholds can be applied to individual interfaces of the multilayer switch to prevent multicast packets with a TTL less than the TTL threshold from being forwarded out the interface. TTL thresholds provide a simple method to prevent the forwarding of multicast traffic beyond the boundary of a site or region, based on the TTL field in a multicast packet. This is known as TTL scoping.

Figure 33-10 shows a multicast packet arriving on Gigabit Ethernet interface 0/2 with a TTL value of 24. Assuming that the RPF check succeeds and that Gigabit Ethernet interfaces 0/1, 0/3, and 0/4 are all in the outgoing interface list, the packet would normally be forwarded out these interfaces. Because some TTL thresholds have been applied to these interfaces, the multilayer switch makes sure that the packet TTL value, which is decremented by 1 to 23, is greater than or equal to the interface TTL threshold before forwarding the packet out the interface. In this example, the packet is forwarded out interfaces 0/1 and 0/4, but not interface 0/3.



Output Interface List
Gigabit Ethernet 0/1: TTL threshold = 16
Gigabit Ethernet 0/3: TTL threshold = 64
Gigabit Ethernet 0/4: TTL threshold = 0

Figure 33-11 shows an example of TTL threshold boundaries being used to limit the forwarding of multicast traffic. Company XYZ has set a TTL threshold of 100 on all routed interfaces at the perimeter of its network. Multicast applications that constrain traffic to within the company's network need to send multicast packets with an initial TTL value set to 99. The engineering and marketing departments have set a TTL threshold of 40 at the perimeter of their networks; therefore, multicast applications running on these networks can prevent their multicast transmissions from leaving their respective networks.



The default TTL value is 0 hops, which means that all multicast packets are forwarded out the interface. The range is 0 to 255.

Only multicast packets with a TTL value greater than the threshold are forwarded out the interface.

You should configure the TTL threshold only on routed interfaces at the perimeter of the network.

```
(config-if)# ip multicast ttl-threshold _value_
```

Configure an IP multicast boundary

Like TTL thresholds, administratively-scoped boundaries can also be used to limit the forwarding of multicast traffic outside of a domain or subdomain. This approach uses a special range of multicast addresses, called administratively-scoped addresses, as the boundary mechanism. If you configure an administratively-scoped boundary on a routed interface, multicast traffic whose multicast group addresses fall in this range can not enter or exit this interface, thereby providing a firewall for multicast traffic in this address range.

Figure 33-12 shows that Company XYZ has an administratively-scoped boundary set for the multicast address range 239.0.0.0/8 on all routed interfaces at the perimeter of its network. This boundary prevents any multicast traffic in the range 239.0.0.0 through 239.255.255.255 from entering or leaving the network. Similarly, the engineering and marketing departments have an administratively-scoped boundary of 239.128.0.0/16 around the perimeter of their networks. This boundary prevents multicast traffic in the range of 239.128.0.0 through 239.128.255.255 from entering or leaving their respective networks.

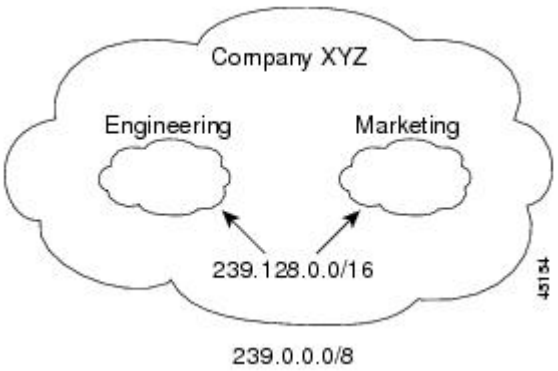


Figure 38. Administratively-Scoped Boundaries

You can define an administratively-scoped boundary on a routed interface for multicast group addresses. A standard access list defines the range of addresses affected. When a boundary is

defined, no multicast data packets are allowed to flow across the boundary from either direction. The boundary allows the same multicast group address to be reused in different administrative domains.

The IANA has designated the multicast address range 239.0.0.0 to 239.255.255.255 as the administratively-scoped addresses. This range of addresses can then be reused in domains administered by different organizations. The addresses would be considered local, not globally unique.

```
(config-if)# ip multicast boundary _standard-access-list-number_
```

Chapter 24. Network optimization

24.1. IP SLA

- **Configuration Guides** | **Network Mgt** | **IP SLA**

To implement IP SLAs network performance measurement,

1. Enable the IP SLAs responder
2. Configure the required IP SLAs operation type
3. Configure any options available
4. Configure threshold conditions
5. Schedule the operation
6. Run the operation for a period of time to gather statistics
7. Display and interpret the results with Cisco CLI or NMS with SNMP

24.1.1. IP SLAs Operation types

- ICMP echo, jitter, path echo , path jitter
- TCP connect
- UDP echo, jitter
- VoIP RTP, UDP jitter, gatekeeper registration delay, post-dial delay
- HTTP, FTP, DNS, DHCP

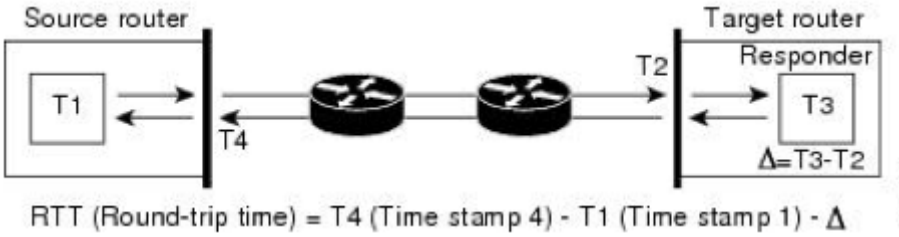
Task: Configure basic IP SLAs ICMP echo operation on the source device

```
(config)# ip sla <operation-number>
(config-ip-sla)# icmp-echo {<destination-ip-address>| <destination-hostname>}
                    [source-ip {<src-ip> | src-hostname} | source-interface
<interface-name>]
(config-ip-sla-echo)# frequency <seconds>
```

24.1.2. IP SLAs Responder

- for Cisco device only
- listens on specific port for control protol messages send by IP SLA operation
- on receipt of the control, open specified TCP or UDP port for specified duration
- disables the port after reponding to the IP SLA packet or specified timeout
- may use MD5 authentication for control message
- takes two timestamps at the target devices to eliminate processing time
- can track one-way delay, jitter, and directional packet loss

- requires NTP synchronization except for one-way jitter measurement



24.1.3. IP SLAs Operation scheduling

Task: schedule IP SLA operation

```
(config)# ip sla schedule <operation-number>
                          [ life {forever} | <seconds> ]
                          [start-time {now | pending | after <hh:mm:ss> |
[<hh:mm:ss>]}] [month day | day month } ]
                          [ageout <seconds>]
                          [recurring]
```

Task: Verify IP SLA configuration

```
# sh ip sla configuration
```

- You can create a multioperation group
  - The frequency of all operations scheduled must be the same
  - The list of operation ID numbers must be limited to a max of 125 chars including commas

Task: schedule a group of IP SLA Operations

```
(config)# ip sla group schedule <group-operation-number> <operation-id-numbers>
                          { schedule-period <schedule-period-range | schedule-together
}
                          [ frequency <group-operation-frequency> ]
                          [ life {forever} | <seconds> ]
                          [start-time {now | pending| after <hh:mm:ss> |
[<hh:mm:ss>]}] [month day | day month } ]
                          [ageout <seconds>]
                          [recurring]
```

Task: Verify IP SLA multioperation configuration

```
# sh ip sla group schedule
```

24.1.4. IP SLAs Operation Threshold Monitoring

- can send SNMP traps that are triggered by connection loss, timeout, round-trip time threshold, average jitter threshold, one-way packet loss, one-way jitter, one-way mean opinion score, one-way latency
- can trigger another IP SLA operation

24.1.5. MPLS VPN awareness

- IP SLA operations can be configured for a specific VPN

24.1.6. History Statistics

Aggregated statistics

By default, two hours of aggregated statistics for each operation. Value from each operation cycle is aggregated with the previously available data within a given hour.

Operation snapshot history

snapshot of data for each operation instance that matches a configurable filter, such as all, over threshold, or failures. The entire dataset is available and no aggregation takes place.

Distribution statistics

frequency distribution over configurable intervals. Each time IP SLAs starts an operation, a new history bucket is created until the number of history buckets matches the specified size or the lifetime of the operation expires. By default, the history for an IP SLAs operation is not collected. If history is collected, each bucket contains one or more history entries from the operation. History buckets do not wrap.

24.1.7. Troubleshooting tips

- If IP SLAs operation is not running and not generating stats, add the **verify-data** command in ip sla configuration mode
- Use **debug ip sla trace** and **debug ip sla error** commands

24.2. Enhanced Object Tracking

- separates definition of tracked object vs action to be taken when tracked object change
- tracked object can be: interface, ip route or ip sla operation or list of objects

24.2.1. Interface Tracking

- can be line protocol or ip route
- may consider the carrier delay timer
- may have a delay between the changes and the notification
- may change the polling frequency

Interface line protocol tracking

Task: Track the line protocol state of an interface

```
(config)# track <object-number> interface <type number> line protocol
```

Interface IP routing tracking

- The IP routing State is up if
  - ip routing is enable and active on the interface
  - known ip address (static, dhcp , ppp/ipcp, unnumbered)
  - interface line protocol up

Task: Track the IP routing state of an interface

```
(config)# track <object-number> interface <type number> ip routing
```

EOT support for carrier delay

- If a link fails, by default there is a two-second timer that must expire before an interface and the associated routes are declared as being down. If a link goes down and comes back up before the carrier delay timer expires, the down state is effectively filtered, and the rest of the software on the switch is not aware that a link-down event occurred. You can extend the timer up to 60 seconds.
- When EOT is configured on an interface, the tracking may detect the interface is down before a configured carrier-delay timer has expired. This is because EOT looks at the interface state and does not consider the carrier delay timer.

Task: (optional) Enables EOT to consider the carrier-delay timer when tracking the status of an interface.

```
(config-track)# carrier-delay
```

Task: (Optional) Specifies a period of time (in seconds) to delay communicating state changes of a tracked object.

```
(config-track)# delay {up <seconds> [down <seconds>] | down <seconds [up <seconds>]}
```

24.2.2. IP route Tracking

- Up if the route exists in the RIB and the route is accessible
- polls the ip route state every 15 seconds

Task: Track an ip route

```
(config)# track <object-number> ip route <a.b.c.d/prefix> reachability
```

Task:(Optional) Specifies the interval in which the tracking process polls the tracked object.

```
(config)# track timer ip route { <seconds> | msec <milliseconds> }
```

TODO: scaled metrics

24.2.3. IP SLA operation tracking

- tracks the state or the reachability of IP SLA operations -

Task: Track the reachability of an IP SLA host

```
(config)# track <object-number> ip sla <operation-number> reachability
```

Example

```
# show track 3
Device# show track 3

Track 3
   IP SLA 1 reachability
   Reachability is Up
     1 change, last change 00:00:47
   Latest operation return code: over threshold
   Latest RTT (millisecs) 4
   Tracked by:
     HSRP Ethernet0/1 3
```

24.2.4. Tracked list

- can be constructed with
  - boolean expression
  - treshold and weight
  - treshold and percentage

Task: Configure tracked list object with a boolean expression

```
(config)# track <list-object-number> list boolean {and | or}
(config-track)# object <object-number> [not]
```

Task: Configure tracked list object with threshold and weight

```
(config)# track <list-object-number> list treshold weight
(config-track)# object <object-number> [weight <number]
(config-track)# object <object-number> [weight <number]
(config-track)# treshold weight {up <number> | down <number>| up <number> down
<number>  }
```

Task: Configure tracked list object with threshold and percentage

```
(config)# track <list-object-number> list treshold percentage
(config-track)# object <object-number>
(config-track)# object <object-number>
(config-track)# treshold percentage {up <number> | down <number>| up <number> down
<number>  }
```

Task: Configure tracked list default

```
(config-track)# default { delay | object <object-number> | treshold percentage }
```

24.3. NetFlow

- Cisco IOS application
- provides statistics on packets flowing through the routing

24.3.1. NetFlow flows

- unidirectional stream of packets between a given source and destination
- combination of 7 fields: source/destination IP address/Port number, layer 3 protocol type, ToS, input logical interface
- can include accounting fields (such as AS number in v5)
- stored/captured in **netflow cache**
- exported to the flow collector (NetFlow Collection Engine)
- prerequisites
  - ip routing
  - cef, dcef or fast switching
  - enough CPU and memory
- data capture
  - from ingress: ip-to-ip, ip-to-mpls, FR and ATM terminated packets
  - for egress: ip-to-ip(with Netflow Accounting), mpls-to-ip (NetFlow MPLS Egress)

Task: Enable NetFlow Capture

```
(config-if)# ip flow { ingress | egress }
```

Task: Verify that NetFlow is operational

```
# sh ip cache [verbose] flow
```

24.3.2. NetFlow version

v5

--

v9

- flexible and extensible
- template based
- uses 2 flow record type: template flowset, and data flowset
- v5 and v9 use the same packet structure

IP Header	UDP Header	NetFlow Header	Flow Record	Flow Record	..	Flow Record

Version 5

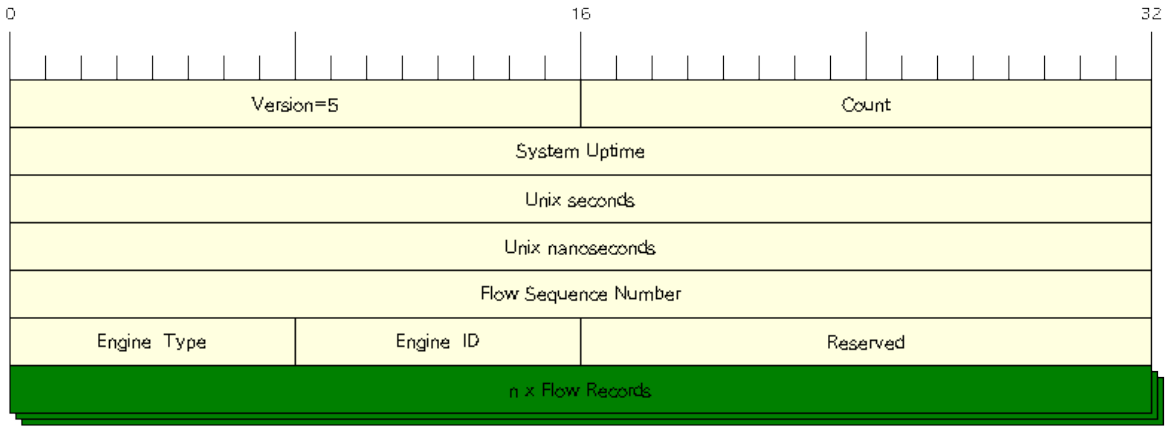


Figure 39. NetFlow header format

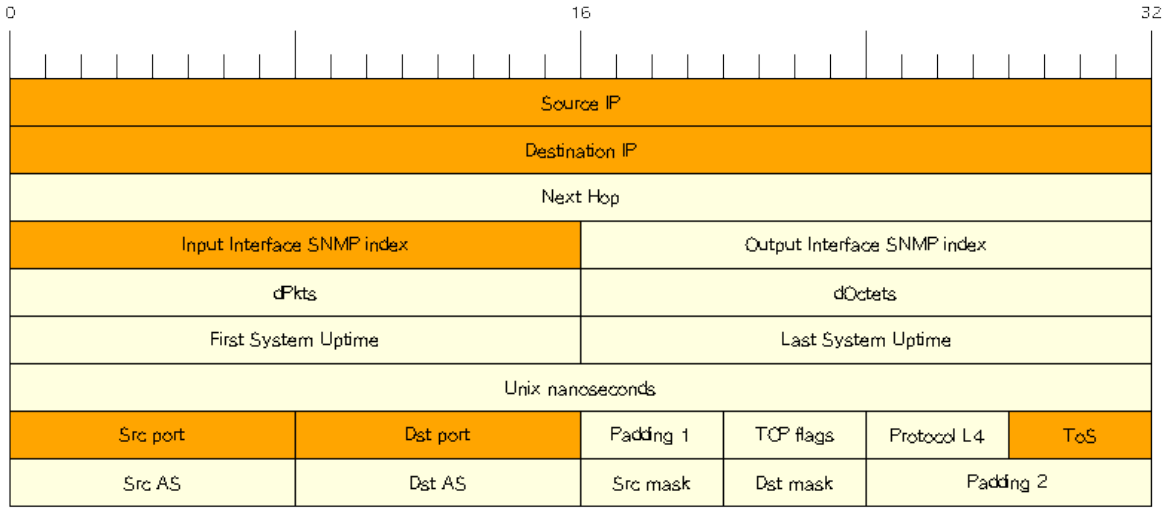


Figure 40. Flow Record format

Version 9

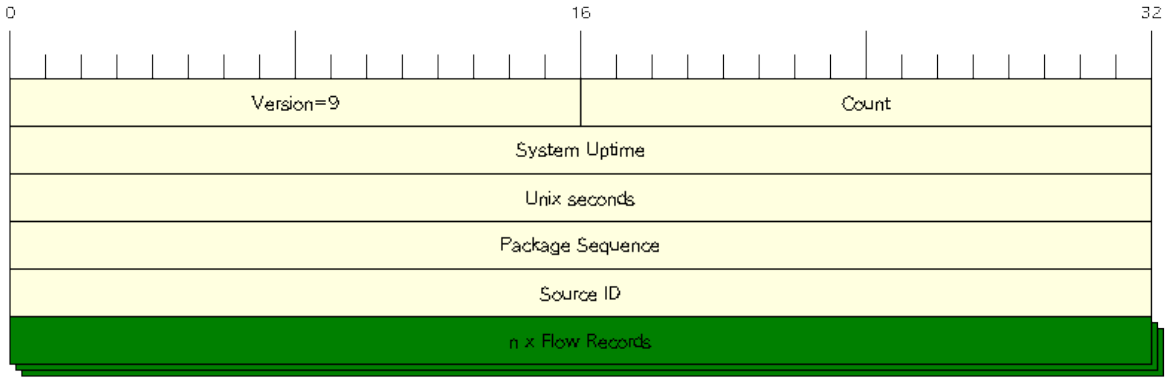


Figure 41. NetFlow header format

24.3.3. NetFlow Cache

- contains flow record for all active flows.
- up to 64K flow entries, each cache entry requires 64 bytes
- removes flows if
  - flow transport is completed (TCP FIN or RST)
  - flow cache full
  - flow becomes inactive after 15 seconds
  - flow active for more than 30 minutes

Task: Configure the size of the NetFlow cache

```
(config)# ip flow-cache entries <size=64000>
```

Task: Configure the flow cache timeout for inactive flow

```
(config)# ip flow-cache timeout inactive <seconds=15>
```

Task: Configure the flow cache timeout for active flows

```
(config)# ip flow-cache timeout active <minutes=30>
```

24.3.4. NetFlow Data Export

- Send NetFlow cache entries to workstation running NetFlow Collection Engine
- supports only two export destinations

Task: Export NetFlow information to a workstation

```
(config)# ip flow-export destination {<ip-address | hostname} <udp-port>
```

Task: (optional) Use version 5 Export Format

```
(config)# ip flow-export version 5
```

Task: (optional) Use version 9 Export Format

```
(config)# ip flow-export version 9
```

Task: verify that NetFlow data export is operational

```
# sh ip flow export
```

24.4. Embedded Event Manager

EEM overview

Part VI : Evolving Technologies

Chapter 25. cloud

IT resources and sces that are abstracted from the underlying infrastructure and provided **on-demand** and **at scale** in a **multitenant** environment.

25.1. Compare and contrast Cloud deployment models

- Public
- Private
- virtual Private
- inter-cloud

25.1.1. Infrastructure, platform, and software services [XaaS]

- SaaS : Application sces
- PaaS: run-time environment and software development frameworks and components presented as API
- IaaS: compute, network and storage
- IT foundation: basic building blocks, core technologies

25.1.2. Performance and reliability

add comparisons table here

25.1.3. Security and privacy

add comparison table here

25.1.4. Scalability and interoperability

add comparison table here

25.2. Describe Cloud implementations and operations

How to connect to the cloud

- Private WAN (like MPLS L3VPN)
- Internet Exchange Point (IXP)
- Internet VPN

25.2.1. Automation and orchestration

Automation: single task Orchestration: process/workflow ordered set of tasks glued together with conditions

25.2.2. Workload mobility

- share workloads accross a collection of resources
- different from VM mobility

25.2.3. Troubleshooting and management

- Scripting languages: Python, Ruby,
- NETCONF
  - transport protocol by which configurations are installed and changed
  - RFC 6241
- YANG
  - modeling language used represent device configuration and state (~ XML)
  - RFC 6020

25.2.4. OpenStack components

TODO: change this to a table

Compute (Nova)

Fabric controller (the main part of an IaaS system). Manages pools of computer resources. A compute resource could be a VM, container, or bare metal server. Side note: Containers are similar to VMs except they share a kernel. They are otherwise independent, like VMs, and are considered a lighter-weight yet secure alternative to VMs.

Networking (Neutron)

Manages networks and IP addresses. Ensures the network is not a bottleneck or otherwise limiting factor in a production environment. This is technology-agnostic network abstraction which allows the user to create custom virtual networks, topologies, etc. For example, virtual network creation includes adding a subnet, gateway address, DNS, etc.

Block Storage (Cinder)

Manages creation, attaching, and detaching of block storage devices to servers. This is not an implementation of storage itself, but provides as API to access that storage. Many storage appliance vendors often have a Cinder plug-in for OpenStack integration; this ultimately abstracts the vendor-specific user interfaces from the management process. Storage volumes can be detached and moved between instances (an interesting form of file transfer, file example) to share information and migrate data between projects.

Identity (Keystone)

Directory service contains users mapped to services they can access. Somewhat similar to group policies applied in corporate deployments. Tenants are stored here which allows them to access resources/services within OpenStack; commonly this is access to the OpenStack Dashboard (Horizon) to manage an OpenStack environment.

Image (Glance)

Provides discovery, registration, and delivery services. These images are like ordinary images to template new virtual services.

Object Storage (Swift)

Storage system with built-in data replication and integrity. Objects and files are written to disk using this interface which manages the I/O details. Scalable and resilient storage for all objects like files, photos, etc. This means the customer doesn't have to deploy a block-storage solution themselves, then manage the storage protocols (iSCSI, NFS, etc).

Dashboard (Horizon)

The GUI for administrators and users to access, provision, and automate resources. The dashboard is based on Python Django framework and is layered on top of service APIs. Logging in relies on Keystone for identity management which secures access to the GUI. The dashboard supports different tenants (business units, groups/teams, customers, etc) with separate permissions and credentials; this is effectively role-based access control. The GUI provides the most basic/common functionality for users without needing CLI access, which is supported for advanced functions. "Security group" abstractions to enforce access control (often need to configure this before being able to access the new instances).

Orchestration (Heat)

Service to orchestrate multiple cloud applications via templates using a variety of APIs.

Workflow (Mistral)

Manages user-created workflows which can be triggered manually or by some event.

Telemetry (Ceilometer)

Provides a Single Point of Contact for billing systems used within the cloud environment.

Database (Trove)

This is a Database-as-a-service provisioning engine.

Elastic Map Reduce (Sahara)

Automated way to provision Hadoop clusters, like a wizard.

Bare Metal (Ironic)

Provisions bare metal machines rather than virtual machines.

Messaging (Zaqar)

Cloud messaging service for Web Developments (full RESTful API) used to communicate between SaaS and mobile applications.

Shared File System (Manila)

Provides an API to manage shares in a vendor agnostic fashion (create, delete, grant/deny access, etc).

DNS (Designate)

Multi-tenant REST API for managing DNS (DNS-as-a-service).

Search (Searchlight)

Provides search capabilities across various cloud services and is being integrated into the Dashboard.

Key Manager (Barbican)

Provides secure storage, provisioning, and management of secrets (passwords).

25.2.5. Resources and References

- http://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/CiscoCloudComputing_WP.pdf
- Open Stack Components
- Cloud Overview
- http://www.unleashingit.com/
- http://www.cisco.com/go/cloud
- https://www.openstack.org/software/
- Designing Networks and Services for the Cloud

Chapter 26. SDN

26.1. Describe functional elements of network programmability and how they interact

SDN: network programmability

- distributed
- augmented
- hybrid
- centralized

26.1.1. Controllers

- responsible for programming forwarding tables of data-plane devices

26.1.2. APIs

26.1.3. Scripting

26.1.4. Agents

26.1.5. Northbound vs. Southbound protocols

26.2. Describe aspects of virtualization and automation in network environments

26.2.1. DevOps methodologies, tools and workflows

26.2.2. Network/application function virtualization [NFV, AFV]

26.2.3. Service function chaining

26.2.4. Performance, availability, and scaling considerations

Chapter 27. Internet of Things

27.1. Describe architectural framework and deployment considerations for Internet of Things [IoT]

27.1.1. Performance, reliability and scalability

27.1.2. Mobility

27.1.3. Security and privacy

27.1.4. Standards and compliance

27.1.5. Migration

27.1.6. Environmental impacts on the network