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

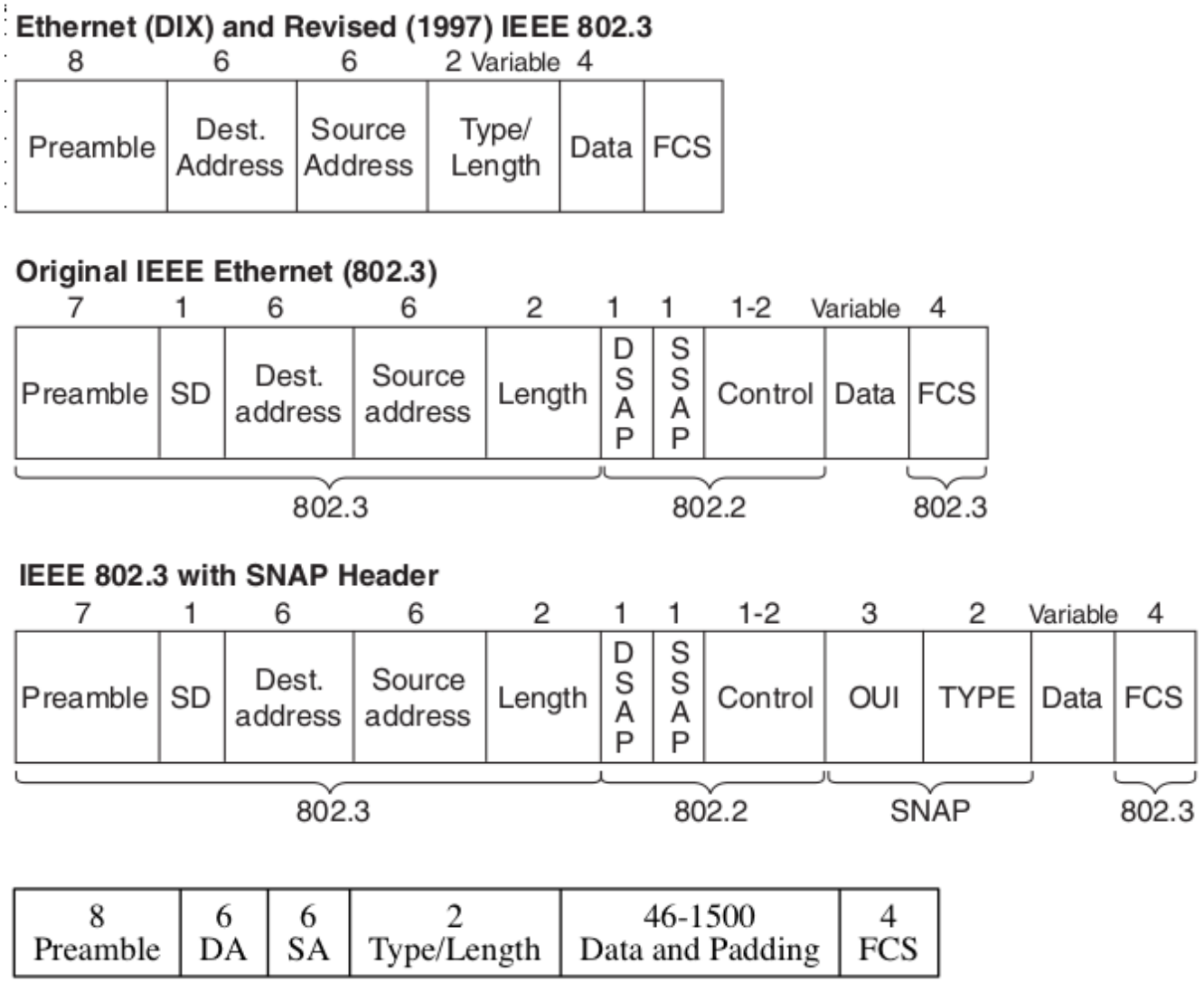


Figure 1. Ethernet (DIX) and Revised (1997) IEEE 802.3

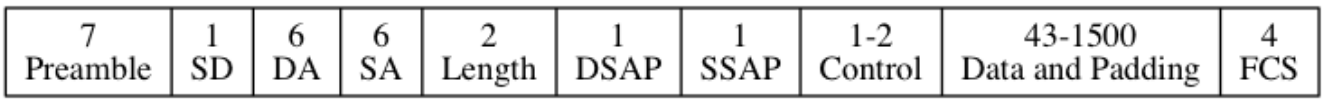


Figure 2. Original IEEE 802.3

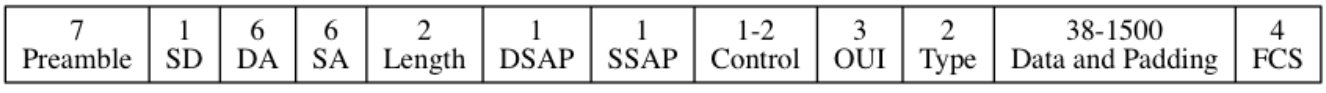


Figure 3. IEEE 802.3 with SNAP

Preamble DIX

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 or protocol header that follows the header.

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 (Organizationally Unique Identifier)

- 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 $46 \Leftarrow n \Leftarrow 1500$
- If $n < 46$, use padding

FCS (Frame check sequence)

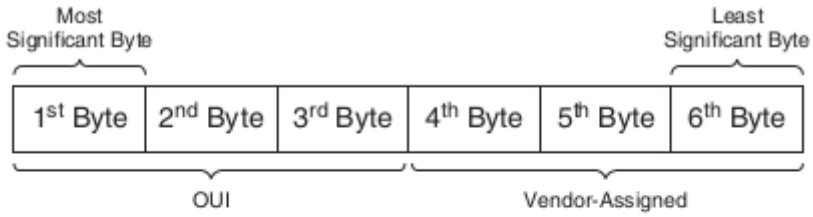
- Contains a 32-bit cyclic redundancy check (CRC) value
- Calculated by the sending MAC
- Re-calculated by the receiving MAC to check for damaged frames.
- Generated from the DA, SA, Length/Type, and Data fields

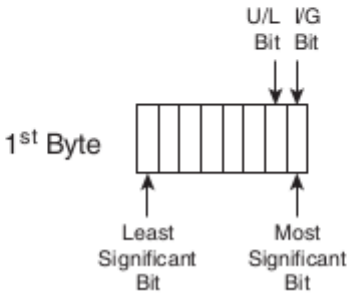
2.2. Ethernet MAC Addresses

- 48 bits in hexadecimal
- Canonical transmission (little endian)= MSO to LSO with LSB to MSB for each octet

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

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





- Fields:

  OUI: first 3 bytes, Organizational Unique Identifier

  I/G: (0/1) individual or Group address, first bit to be transmitted

  U/L: (0/1) universally or Locally administrated

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

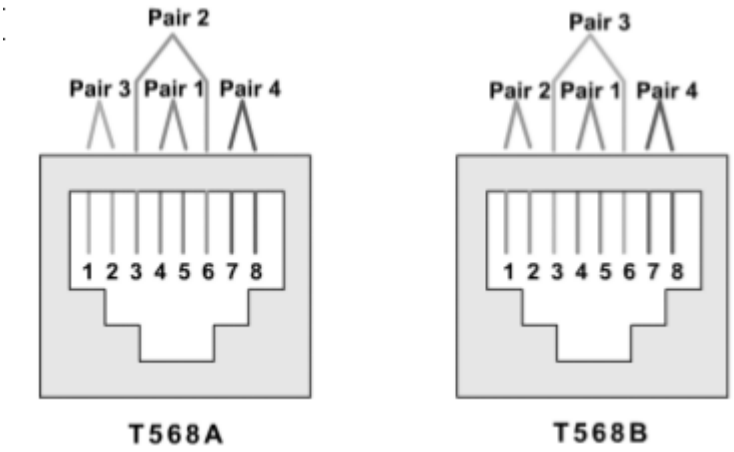


Table 1. Ethernet cabling types

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, which solves the cabling problem.
  - Not supported on all Cisco switch models.

2.4. Auto-negotiation, Speed and Duplex

By default, each Cisco switch port uses Ethernet auto-negotiation to determine the speed and duplex setting.

Switches can dynamically detect the speed setting on a particular Ethernet segment by using a few different methods. Cisco switches (and many other devices) can sense the speed using the Fast Link Pulses (FLP) of the auto-negotiation process. However, if auto-negotiation is disabled on either end of the cable, the switch detects the speed anyway based on the incoming electrical signal. You can force a speed mismatch by statically configuring different speeds on either end of the cable, causing the link to no longer function.

Task: Set speed for the interface

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

Switches detect duplex settings through auto-negotiation only. If both ends have auto- negotiation enabled, the duplex is negotiated. However, if either device on the cable disables auto-negotiation, the devices without a configured duplex setting must assume a default. Cisco switches use a default

duplex setting of half duplex (HDX) (for 10-Mbps and 100-Mbps interfaces) or full duplex (FDX) (for 1000-Mbps interfaces). To disable auto-negotiation on a Cisco switch port, you simply need to statically configure the speed and the duplex settings. Ethernet devices can use FDX only when collisions cannot occur on the attached cable; a collision-free link can be guaranteed only when a shared hub is not in use. The next few topics review how Ethernet deals with collisions when they do occur, as well as what is different with Ethernet logic in cases where collisions cannot occur and FDX is allowed.

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

Task: Show MAC address table

```
Switch1# show mac-address-table dynamic
  Mac Address Table
  ------------------
  Vlan Mac Address Type Ports
  ---- ----------- ---- -----
  1 000f.2343.87cd DYNAMIC Fa0/13
  1 0200.3333.3333 DYNAMIC Fa0/3
  1 0200.4444.4444 DYNAMIC Fa0/13
```

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

2.8. Troubleshooting

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

Chapter 3. CDP

Catalyst3560-X Configuration Guides | **CDP**

3.1. Overview

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

3.1.1. Packet format

- Header followed by a set of TLV value

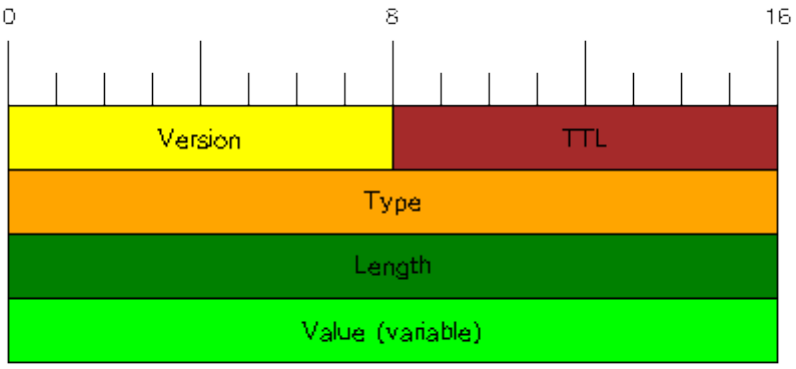


Figure 4. CDP frame format

3.2. 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
```

3.2.1. 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

3.2.2. Version

Task: Send Version-2 advertisements

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

3.3. Monitoring and maintenance

Task: Reset the traffic counters to zero

```
clear cdp counters
```

3.3.1. 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
```

Chapter 4. LLDP

Catalyst Configuration Guides | **LLDP**

4.1. 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

4.2. 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
```

4.3. 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>]
```

4.4. 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]
```

4.5. 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
```

4.6. 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
```

4.7. Network-policy profiles

Task: Display the configured network-policy profiles.

```
show network-policy profile
```

Task: Display the NMSP information.

```
show nmsp
```

4.8. 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

4.9. 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.

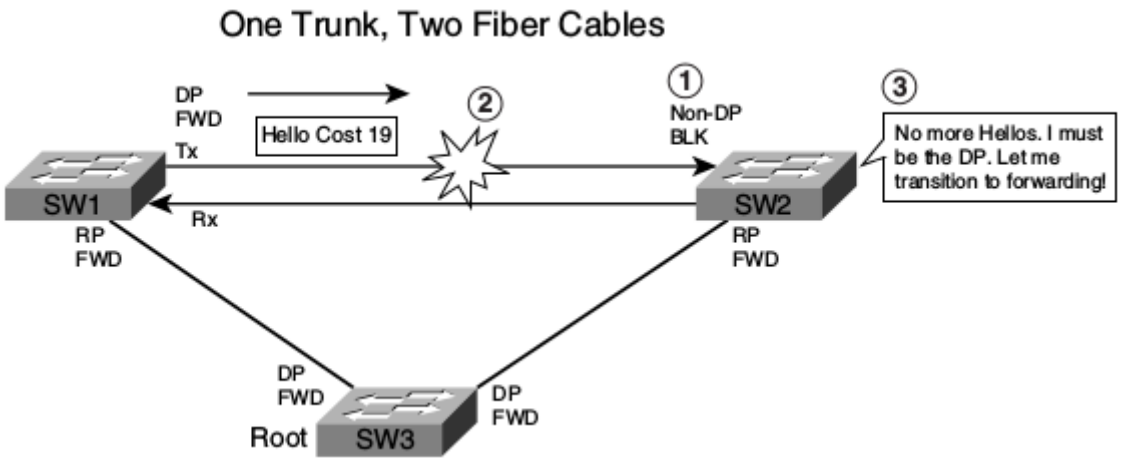
If you change a location address on the switch, the switch sends an NMSP location notification message that identifies the affected ports and the changed address information.

Chapter 5. UDLD

Catalyst 3560 Configuration Guides | **UDLD**

5.1. 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.

5.1.1. Modes of operations

Normal

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

Aggresive mode

5.2. Tasks

5.2.1. 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]**

5.3. 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 6. VLAN

6.1. VLAN

6.1.1. concepts

- 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"

Private VLANs

- separate ports as if they are on different VLAN while consuming only one subset.
- typically use by service provider in a multi-tenant offerings: one router, one switch, multiple customers
- pVLAN = one primary VLAN (promiscuous ports) + multiple secondary VLANs (community and isolated ports)

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

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

6.1.2. Configuration tasks

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

6.1.3. 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

6.2. Voice VLAN

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

6.3. Private VLANs

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

Chapter 7. Trunking

7.1. VTP

Catalyst Configuration guides | **Configuring VTP**

7.1.1. Overview

- 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)

7.1.2. 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

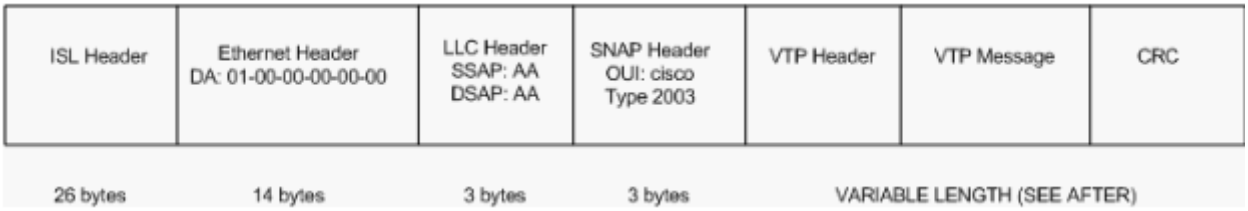


Figure 5. Example: VTP packet encapsulated in ISL frame

- 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

- Informs adjacent Catalysts of the current VTP domain name and the configuration revision number.
- 5 minutes intervals
- 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.

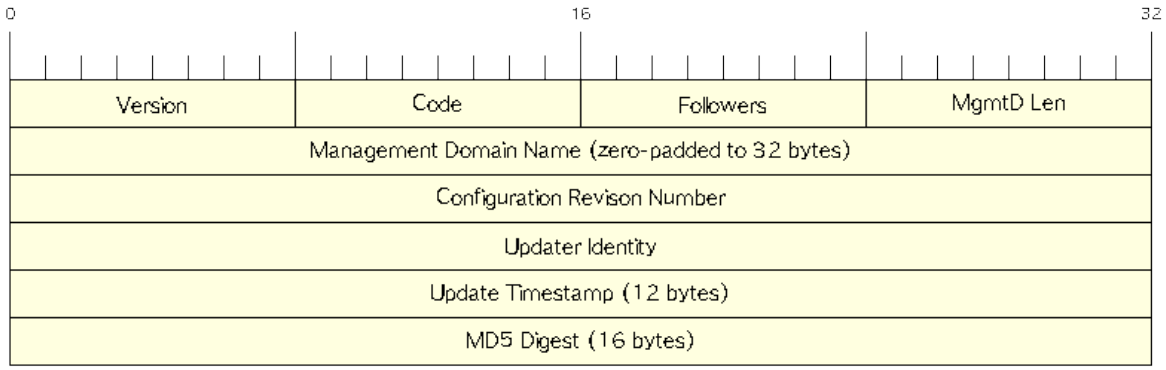


Figure 6. VTP Summary Advert

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.

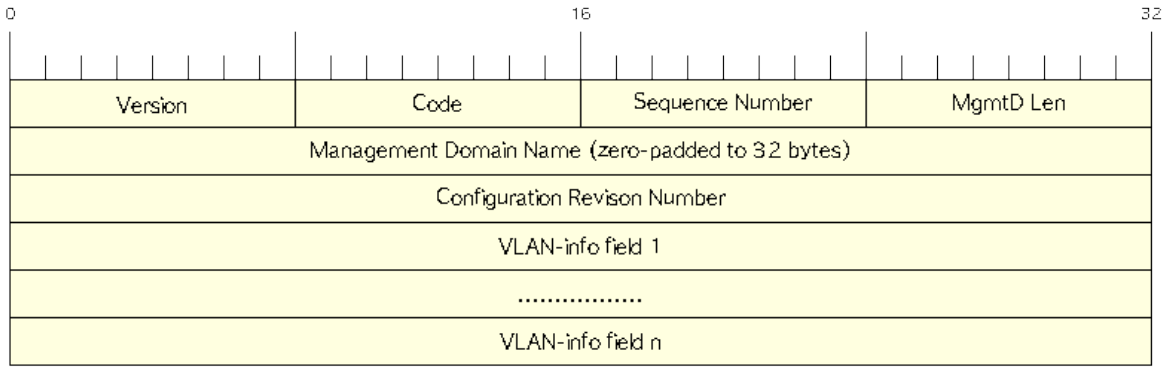


Figure 7. VTP Subset advertisements

Code

value: 0x02 for subset advertisement.

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:

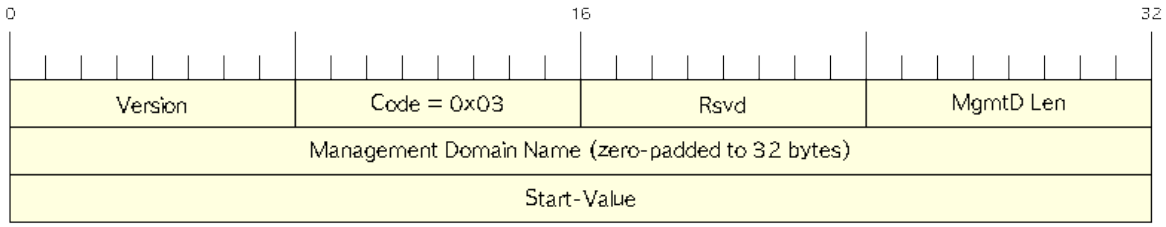


Figure 8. VTP advertisement request

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.

7.1.3. VTP domains

- 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>
```

7.1.4. Configuration revision number

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

7.1.5. Allowed, active and pruned VLANs

Although a trunk can support VLANs 1–4094, several mechanisms reduce the actual number of VLANs whose traffic flows over the trunk. First, VLANs can be administratively forbidden from existing over the trunk using the **switchport trunk allowed** interface subcommand. Also, any allowed VLANs must be configured on the switch before they are considered active on the trunk. Finally, VTP can prune VLANs from the trunk, with the switch simply ceasing to forward frames from that VLAN over the trunk.

The **show interface trunk** command lists the VLANs that fall into each category:

Allowed VLANs

Each trunk allows all VLANs by default. However, VLANs can be removed or added to the list of allowed VLANs by using the switchport trunk allowed command.

Allowed and active

To be active, a VLAN must be in the allowed list for the trunk (based on trunk configuration), and the VLAN must exist in the VLAN configuration on the switch. With PVST+, an STP instance is actively running on this trunk for the VLANs in this list.

Active and not pruned

This list is a subset of the “allowed and active” list, with any VTP-pruned VLANs removed.

7.1.6. 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

```
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

```
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

```
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

7.1.7. 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

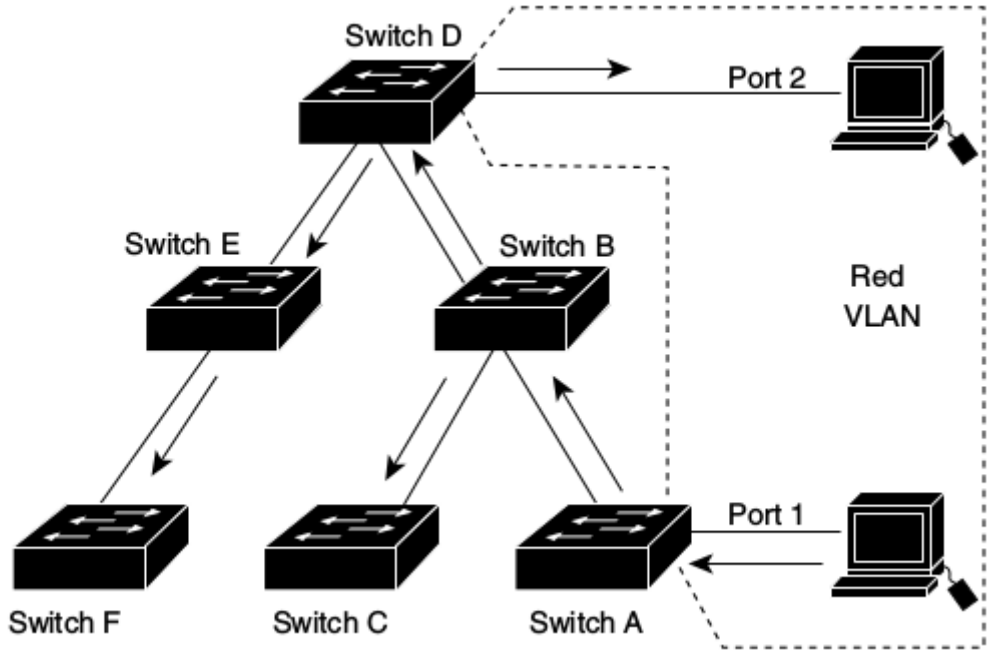
```
(config)# vtp password <string>
```

Task: Show the VTP password

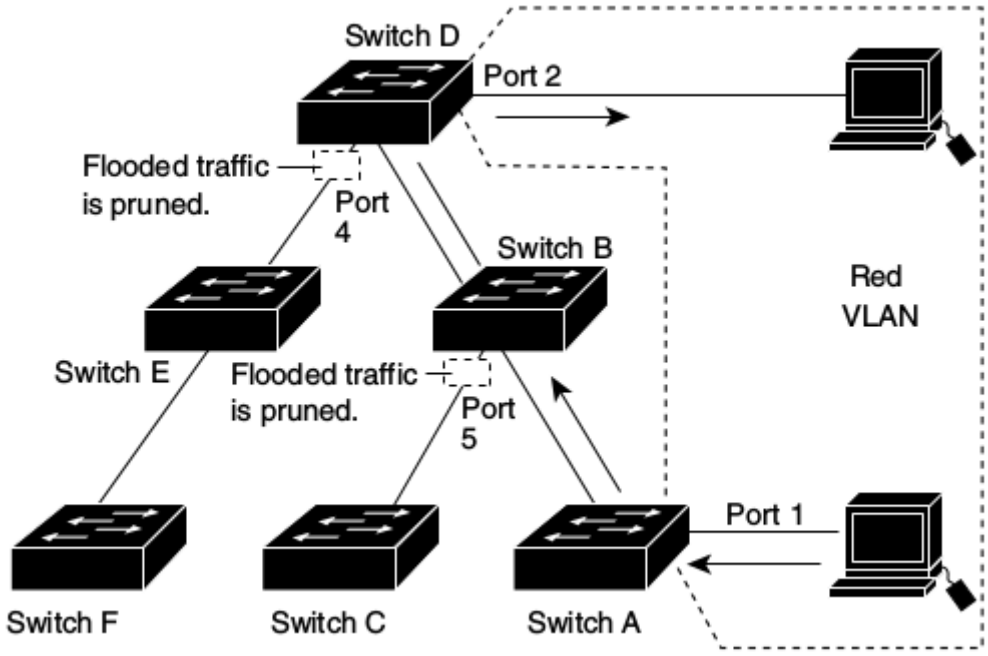
```
(config)# sh vtp password
```

7.1.8. 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.

7.1.9. Version

TODO: Add task for this section

- Default: version 1

Version 2

-

Version 3

- 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

For more information, read VTP version 3

Task: Verify VTP configuration

```
# 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
```

7.1.10. Troubleshooting

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

7.2. DTP

7.2.1. Dynamic trunk protocol

- negotiate trunk status
- default to **dynamic auto**

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

Configuration Command	Short name	Meaning	To trunk other side must be
switchport mode access; switchport nonegotiate	Access (with nonegotiate)	Never trunks; does not send DTP messages	(Never trunks)

Task: Configure an inter-switch links 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>
```

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 123
```

7.2.2. Verify

What is TOS/TAT in

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

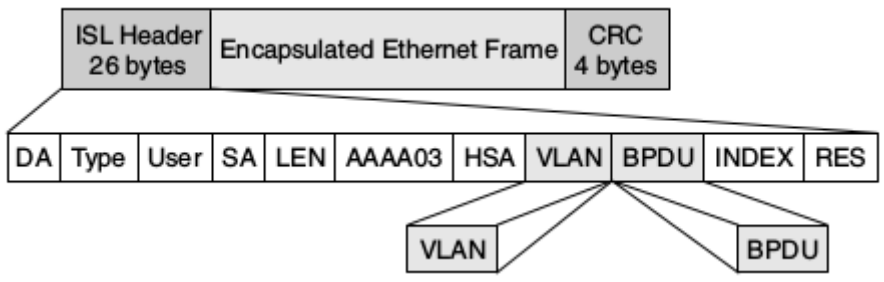
7.3. ISL

7.3.1. Overview

- Inter-Switch Link
- Cisco proprietary
- Provides VLAN trunking
- Supports up to 1000 VLANs (check ???)
- Encapsulates the original header with 26-byte header
- Removes the header at the receiving end

7.3.2. 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

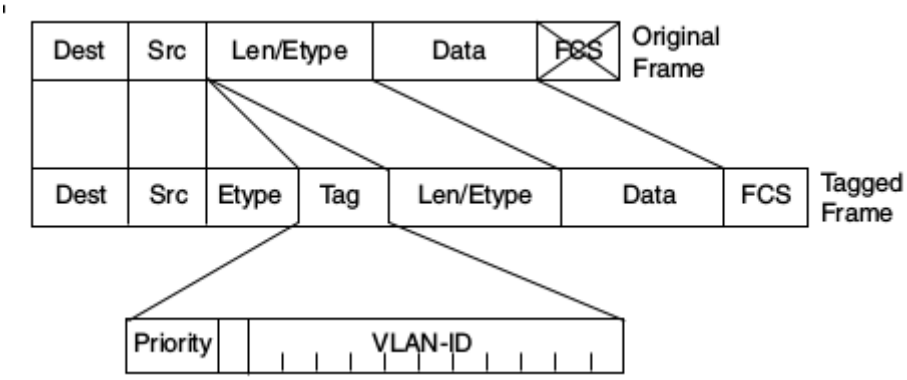
7.4. IEEE 802.1Q

7.4.1. 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.

7.4.2. 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

7.4.3. Native VLAN

Task: Configure a native VLAN over a trunk link

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

Chapter 8. Spanning tree

8.1. 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

8.1.1. 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

Root port election

- RP is upstream facing towards Root bridge
- lowest root path cost (cumulative cost of all links to get to the root)

- cost based on inverse bandwith

Table 5. default port costs

Speed	original	revised
10 Mbps	100	100
100 Mbps	10	19
1 Gbps	1	4
10 Gbps	1	2

Tie breaker when a switch receives multiple Hellos with equal cost

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

Task: Force election of a root bridge

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

determining the 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 all non-RP and non-DP 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.

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	stable

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 sd 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

configuration

show spanning-tree root show 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

```
(cfg-if) spanning-tree portfast
(cfg) 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.

```
(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

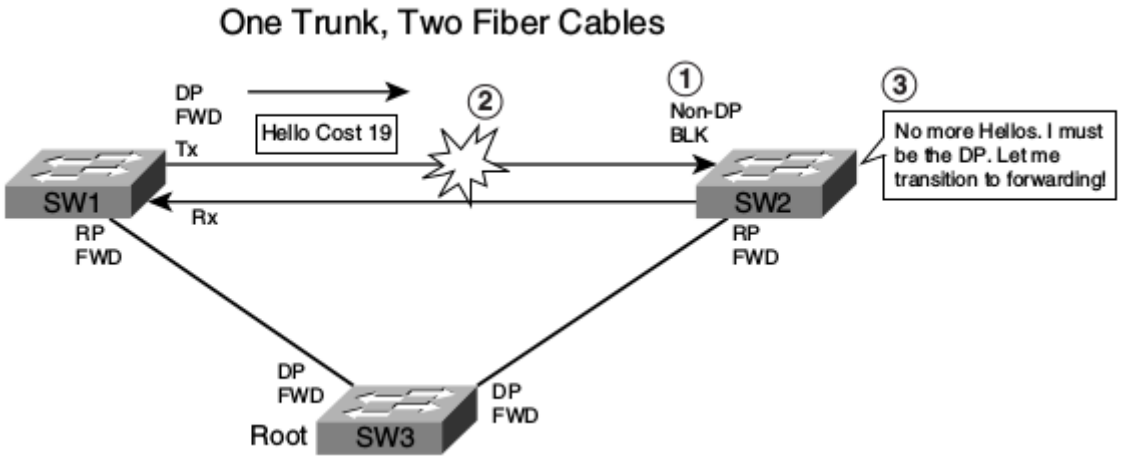
- Prevent a port from becoming a root port when receiving a superior bpdu (e.g. inferior priority + mac)
- it is enabled on ports other than the root port and on switches other than the root.
- Puts the port in **root-inconsistent** state (no data flow) until it stops receiving superior BPDUs. No traffic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

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.

  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.

8.1.2. 802.1w

- 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 although
- All bridges generate BPDUs every Hello interval

RSTP link types

- **Point-to-point**: switch to switch
- **Shared** : switch to hub
- **Edge**: switch to single end-user device

RSTP port states

administrative state	**802.1d**	**802.1w**
disabled	disabled	discarding
enabled	blocking	discarding
enabled	listening	discarding
enabled	learning	learning
enabled	forwarding	forwarding

RSTP role ports

Root Port

- Same role as 802.1d RP

Designated Port

- Same role as 802.1d DP

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 features
- protects against losing the DP attached to a shared liknk when the switch has another physical port attached to the same shared segment

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

a port needs to receive BPDUs to stay blocked.

configuration

Task: Configure Rapid PVST

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

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

8.1.3. 802.1s

- 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 ???

storm control

unicast flooding

8.1.4. Troubleshooting

flapping port that is generating BPDUs with the TCN bit set

8.1.5. Questions

8.2. MST

8.2.1. operations

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

8.2.2. 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 9. EtherChannel

9.1. EtherChannel

9.1.1. Overview

- 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
- Channel 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.

9.1.2. 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.

9.1.3. 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]  }
```

9.1.4. 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

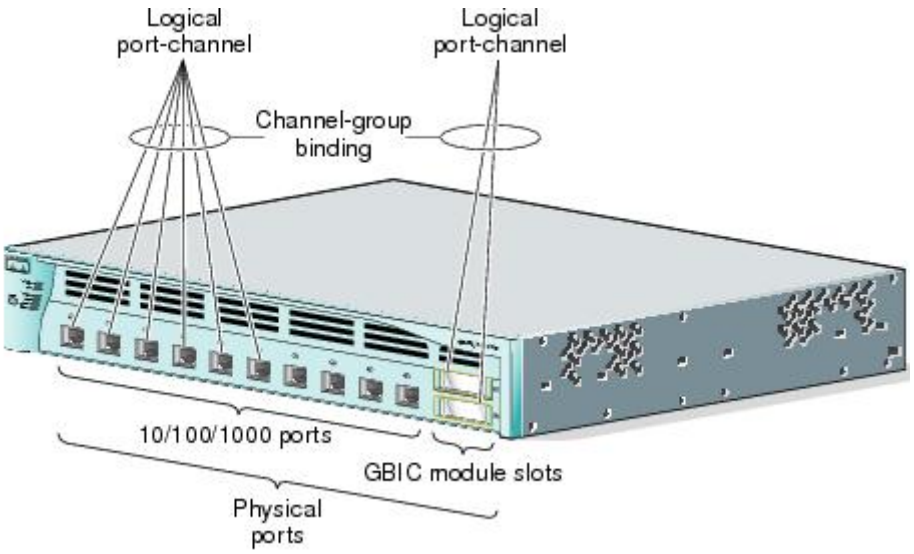


Figure 9. Relationship of Physical Ports, Logical Port Channels, and Channel Groups

9.1.5. EtherChannel modes

Table 6. 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
```

9.1.6. Misconfiguration guard

TODO

9.2. LACP

9.2.1. 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

9.2.2. 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
```

9.2.3. 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
```

9.2.4. 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
```

9.3. PAgP

9.3.1. 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]
```

9.3.2. 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.

9.3.3. 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
```

9.3.4. 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>
```

9.3.5. 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.

9.3.6. 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.

9.3.7. 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 10. Monitoring

10.1. SPAN

Chapter 11. Multicast

11.1. IGMP

Configuration guides | **Multicast** | **IGMP**

Check also Catalyst configuration guides

11.1.1. 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

11.1.2. 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

11.1.3. 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>
```

11.1.4. 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
```

11.1.5. 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

11.1.6. 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>
```

11.1.7. IGMP proxy

TODO

11.1.8. 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 7. 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
```

11.1.9. 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 8. 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]
```

11.1.10. IGMP filtering and throttling

Table 9. 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 }
```

11.2. PIM

11.2.1. 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).

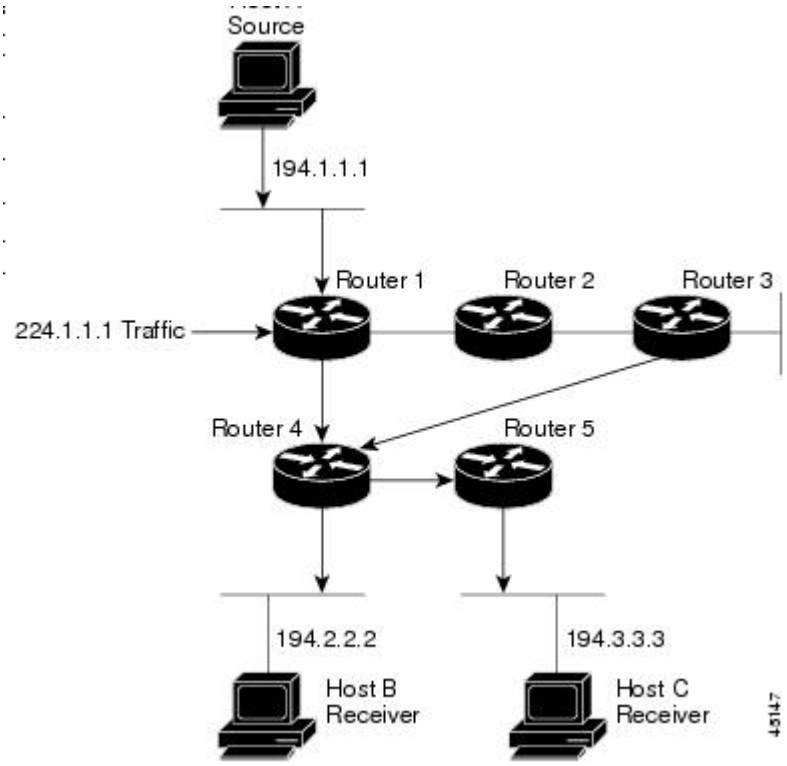


Figure 10. Host A Shortest-Path Tree

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.

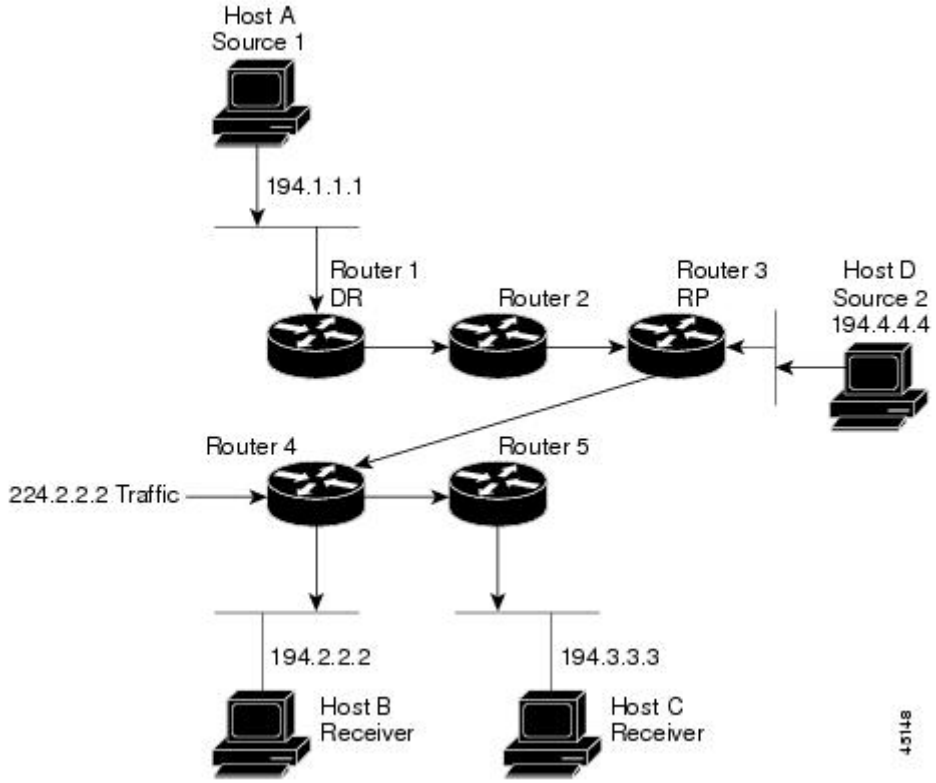


Figure 11. Shared Distribution Tree

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.

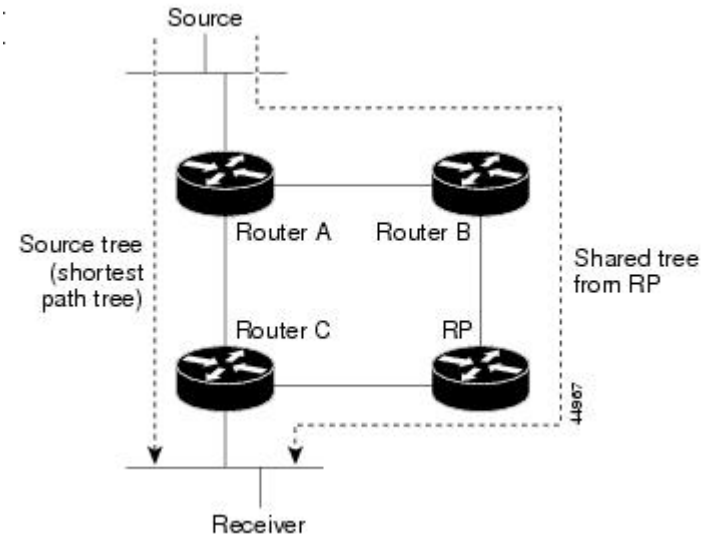


Figure 12. PIM trees

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:

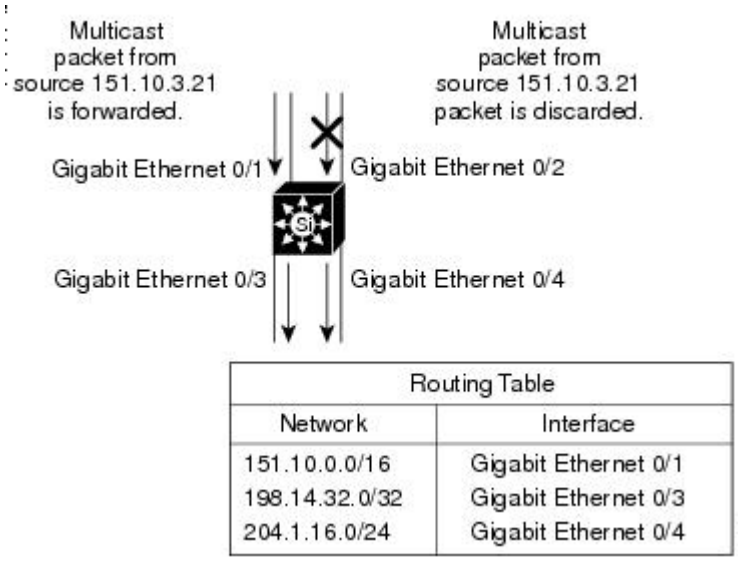
1. The router examines the source address of the arriving multicast packet to determine whether the packet arrived on an interface that is on the reverse path back to the source.
2. If the packet arrives on the interface leading back to the source, the RPF check is successful and

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 13. 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.

11.2.2. 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
```

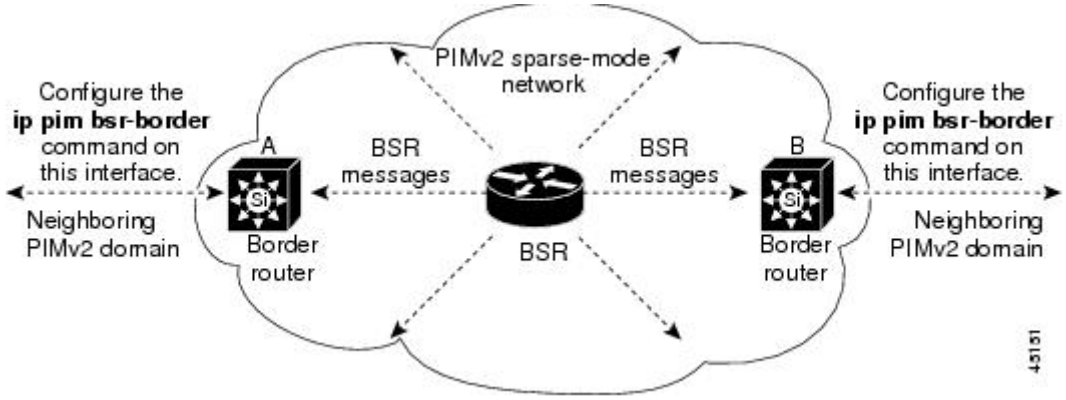


Figure 14. Constraining PIMv2 BSR Messages

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]
```

11.2.3. 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

11.2.4. Misc

TODO To be added in the text

Table 10. 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}
```

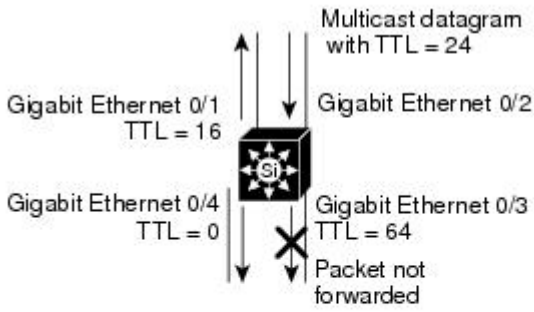
default defines the ssm range access list to 232/8

TODO: Need to delete section below.

Configure the TTL Threshold

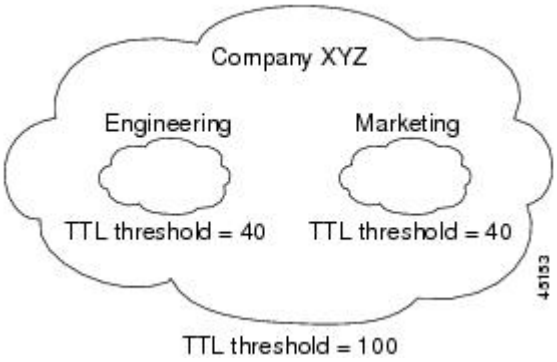
Each time an IP multicast packet is forwarded by the multilayer switch, the time-to-live (TTL) value in the IP header is decremented by one. If the packet TTL decrements to zero, the switch drops the packet. TTL thresholds can be applied to individual interfaces of the multilayer switch to prevent multicast 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.

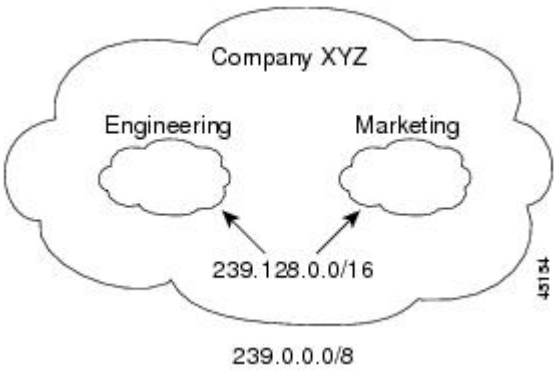


Figure 15. Administratively-Scoped Boundaries

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 12. WAN

12.1. HDLC

12.1.1. Concepts

12.1.2. Configuration

12.2. PPP

12.2.1. Concepts

- HDLC (high-level data link control) and PPP: layer 2 on point-to-point links
- ISO HDLC does not include a Type field , so the Cisco implementation adds a proprietary 2-byte Type field
- hdlc : error detection, default on IOS serial links
- ppp : error detection, error recovery, standard protocol Type field, supports synchronous and asynchronous links
- hdlc vs ppp framing

PPP LCP

- LCP (link control protocol) controls features independent of any Layer 3 protocol
- NCP (network control protocol) for each protocol (IP, appletalk,)
- LCP operations:
- 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

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

- 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:

configuration

ppp multilink interleave ppp multilink fragment-delay ms defines the fragment size based on size = x * bandwidth

ppp compression

- use 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

12.2.2. Configuration tasks

```
debug ppp authentication
```

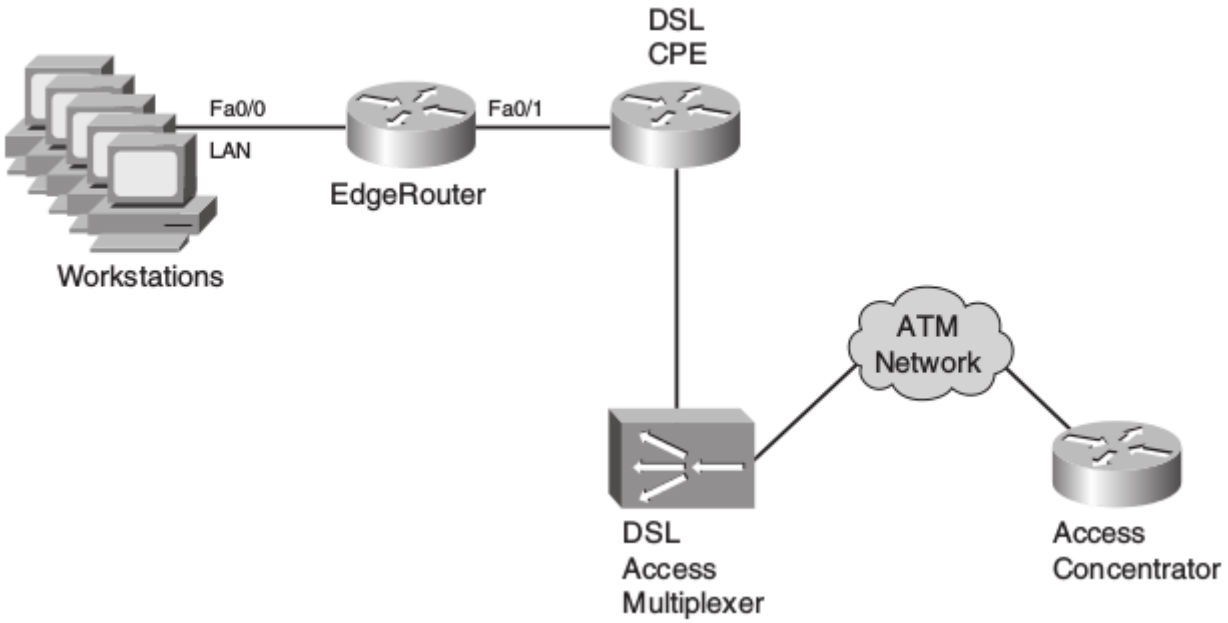
read also understanding debug ppp negotiation

12.2.3. 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.

12.2.4. Configuration tasks

Example



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
```

Verify PPPoE connectivity

```
show pppoe session
```

Debug

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

Part II : Layer 3 Technologies

Chapter 13. IPv4

13.1. Concepts

- RFC 791

13.1.1. IP packet format

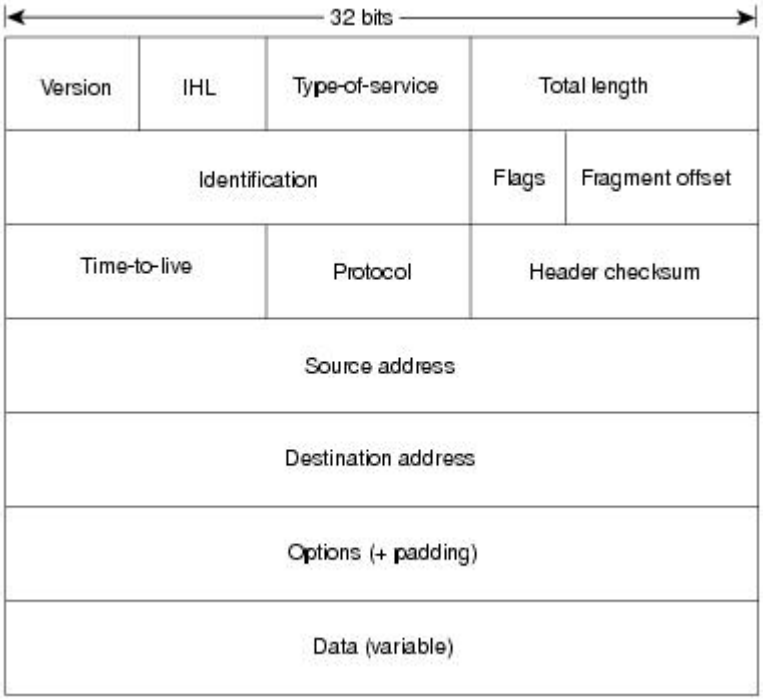


Figure 16. IP packet format

Version

Indicates the version of IP currently used.

IP Header Length (IHL)

Indicates the 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.

Source Address

Specifies the sending node.

Destination Address

Specifies the receiving node.

Options

Allows IP to support various options, such as security.

Data

Contains upper-layer information.

13.1.2. IP addressing

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

13.1.3. 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

13.1.4. Private addressing

- RFC 1918

10.0.0.0/8

- 172.16.0.0/12
- 192.168.0.0/16

13.1.5. VLSM

- Variable length subnet mask

13.2. Configuration tasks

13.2.1. Assign an IP address to an interface

```
(config-if)# ip address ip-address mask [secondary]
```

13.2.2. Allow the use of IP subnet zero

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

13.2.3. Specify the format in which netmask appear for the current session

```
term ip netmask-format {bitcount | decimal | hexadecimal}
```

13.2.4. Specify the format in which netmask appear for the current line

Enters line configuration mode for the range of lines specified by the first and last arguments.

```
(config)# line vty first last
```

Specifies the format the router uses to display the network mask for an individual line.

```
term ip netmask-format {bitcount | decimal | hexadecimal}
```

13.2.5. Use IP unnumbered interfaces on point-to-point WAN interfaces

- Borrow the IP address of another interface

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

Restrictions

- only point-to-point (non-multiaccess) WAN interfaces

You cannot reboot a IOS image over an ip unnumbered interface

13.2.6. Use a 31-bit prefix on point-to-point WAN interfaces

- Since RFC 3021
- only on point-to-point WAN interfaces

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

13.3. Troubleshooting

13.3.1. Display the IP parameters for the interface

```
show ip interface
```

13.3.2. Display the IP networks the device is connected to

```
show ip route connected
```

13.3.3. RFC 5227 - IPv4 address conflict detection

Chapter 14. GRE

14.1. Concepts

14.1.1. Tunneling

- Tunneling encapsulates data packets 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.

Components:

- **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.

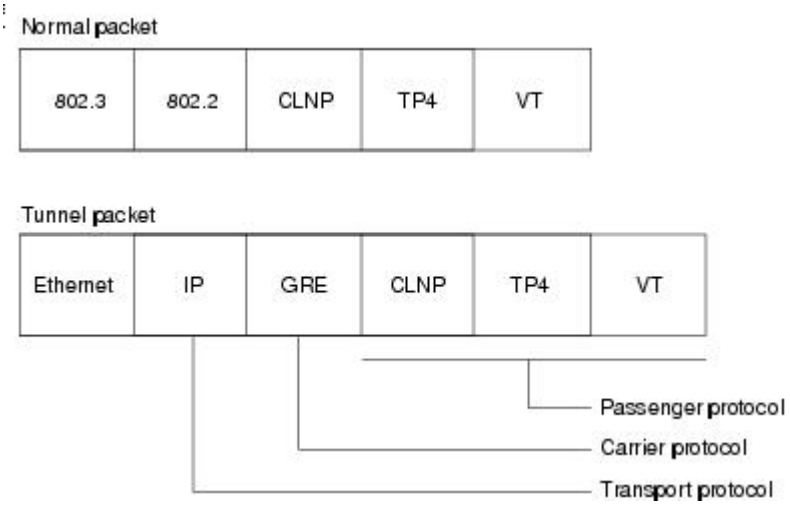
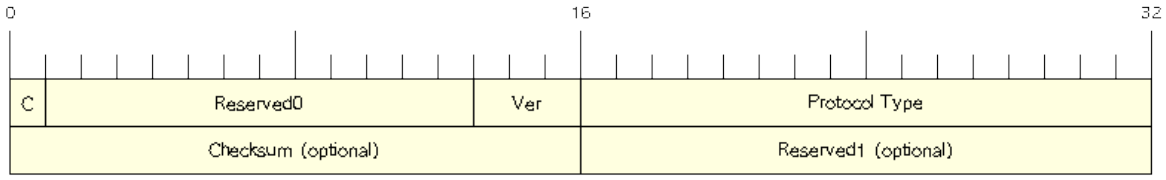


Figure 17. IP Tunneling Terminology and Concepts

14.1.2. GRE

- IP protocol 47
- RFC 2784
- RFC 2345
- RFC 1234

GRE header



Checksum present

- bit 0
- indicates that the checksum and the Reserved1 field are present

Reserved0

- 12 bits
- if any of bits 1-5 are non-zero, a receiver must discard the packet unless receiver implements RFC1701

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.

14.2. Configuration

14.2.1. Configure a 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.

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| gre multipoint}
  ip mtu bytes
  ip tcp mss mss-value
  tunnel path-mtu-discovery [age-timer {aging-mins| infinite}]
```

Create a tunnel interface

```
interface tunnel number
```

Specify a source interface for the tunnel.

The tunnel source interface can be a local physical or logical local interface, and not just an IP address

```
tunnel source {a.b.c.d | source-interface }
```

Specify the destination IP address for the tunnel

The router should have a route to this address, but not through the tunnel interface.

```
tunnel destination ip-address
```

Specify the tunnel mode

The default tunnel mode is **gre ip**.

```
tunnel mode [gre {ip | multipoint} | dvmrp | ipip | mpls | nos]
```

Adjust the GRE keepalive

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.

```
(config-if)# keepalive [ period [retries]]
```

14.2.2. 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
```

14.3. 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.

14.3.1. "%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

14.4. Questions

1. What is the minimum amount of additional header that GRE adds to a packet?
    a. 16 bytes
    b. 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

Chapter 15. RIP

Configuration Guides | IP Routing | RIP

15.1. Overview

- 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:
  - 1 to 6
  - 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.
- Support CIDR, VLSM, authentication
- Periodic updates every 30 seconds to multicast address 224.0.0.9
- Split horizon with poison reverse
- 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
- Does not keep a separate topology table

15.2. Default RIP configuration

- version : 1
- auto-summary : enable
- authentication : disable
- authentication mode: text
- split-horizon : enable

- Interpacket delay : no

15.3. Basic configuration

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

15.4. 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}
```

15.5. Authentication

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

15.6. 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>
```

15.7. 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
```

15.8. Route filtering

Task: Stop advertising a route with a prefix-list

```

```

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

15.9. Route metric

15.10. Split horizon

Task: Disable split horizon

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

15.11. 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>
```

15.12. Rip Optimization over WAN

Task: Enable triggered extensions for RIP

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

15.13. Offset-list

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

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

15.14. Timers

Task: Configure RIP timers

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

- Update timer: interval between updates. Default: 30 seconds
- Invalid timer: time in seconds after which a route is declared invalid.
  - Should be at least 3 times the update timer.
  - Invalid routes are still used for forwarding packets
  - Default: 180 seconds
- Holdown timer: interval during which routing information about better paths is suppressed.
  - 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
  - Default: 180 seconds
- Flush timer: amount of time that must pass berore a route is removed from the RIB. Default: 240 seconds
- 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 comd above overrides the update timers set by **timers basic** command.

Chapter 16. EIGRP

16.1. Overview

- 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
- Forms active neighbor adjacencies
- DUAL for loop-free topology and fast convergence
- granular metric
- unequal cost load balancing
- summarization
- Supports MD5 based authentication

16.2. EIGRP messages

Hello

- multicast to 224.0.0.10
- do not require acknowlegment
- can be used as Ack if sent without data

Ack

- unicast
- contains a nonzero acknowledgement number

Update

- multicast or unicast

Query

- multicast unless in response to a received query

Reply

- 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

Task: Exchange EIGRP packets only as unicast

```
(config-router)# neighbor <a.b.c.d> <interface-id>
```

Task: Exchange EIGRP packets only as unicast

```
(config-router-af-interface)# neighbor <a.b.c.d> <interface-id>
```

Task: debug EIGRP

```
debug ip eigrp packet [hello | ack | update } quey | reply]
```

16.3. Neighbors

- discovered with Hello packets
- must agree on
- primary IPv4 subnet
- Autonomous System Number
- authentication
- K values
- do not need to agree on timers

Task: Verify neighbor adjacencies

```
# sh ip eigrp neighbors [detail]
```

Check that the queue count is zero

16.4. EIGRP Loop prevention techniques

16.5. Split horizon

- Enable 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
```

16.6. DUAL feasibility condition

16.7. EIGRP reconvergence

- if the successor becomes unreachable,
  - if there is a feasible successor
    - use the FS without local computation
  - else
    - query the neighbors and put the route in active
- Transport: IP protocol 88
- Administrative distance : 90 internal routes, 5 summary routes , 170 external routes
- Hello interval

16.8. Metric

metric = 256 * [K1 x bandwidth + (k2 x bandwidth)/ (256 -load) + k3 x delay] x [k5 /(reliability + k4)]

- bandwith: 10^7 / minimum bandwidth in Kbps
- delay: in tens-of-microseconds
- reliability: likelihood of successful packet transmission with 0 means 0% and 255 means 100%
- load : effective load of the route with 255 means 100% loading
- mtu : minimum Maximum transmission unit
- default values: k1,k2,k3,k4,k5 = 1,0,1,0,0
- the values of K must match for the neighbors to become adjacents

Task: Description

```
(config-router)# metric weights
```

16.9. Wide Metric

Metric = [(K1*Minimum Throughput + (K2*Minimum Throughput/(256-Load) + (K3*Total Latency) + (K6*Extended Attributes)]* [K5/(K4 + Reliability)]

16.10. 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.

16.11. 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
```

16.12. EIGRPv3

```
(config-router)# address-family ipv6 [unicast] [vrf vrf-name] autonomous-system asn
```

16.13. EIGRP Neighbor Relationship Maintenance

- Hellos
- adjancency

16.14. DUAL Finite State Machine

- a successor is a neighboring router that has a least-cost path to a destination that is guarentedd not to be part of a routinf

- Feasibility condition: RD < FD

16.15. Protocol-Dependent Modules

16.16. Goodbye Message

- broadcast when an EIGRP routing process is shut down
- Speeds convergence as peers don't have to wait the hold timer expiration
- 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

16.17. Routing Metric Offset Lists

16.18. EIGRP Cost Metrics

16.19. Summarization

- All subnets are suppressed

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

16.19.1. Leak map

Task: Advertise specific subnets of a EIGRP summary

```
(config-if)# ip summary-address eigrp <asn> <prefix> <mask> leak-map <route-maps>
```

16.19.2. Floating Summary Routes

TODO - By default, summarization install a route to Null0 to match the summary to prevent forwarding traffic for unreachable destinations. -

16.19.3. Poisoned Floating Summarization

TODO

16.20. 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

16.21. Hello Packets and the Hold-Time Intervals

16.22. Split Horizon

16.23. 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

16.24. EIGRP Stub Routing

16.25. EIGRP Stub Routing Leak Map Support

16.26. 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

16.27. verify eigrp topology

```
show ip eigrp topology [all-links]
show ip eigrp topology [prefix/len]
```

Chapter 17. OSPF

Configuration Guides | IP Routing | OSPF

17.1. Overview

- 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

17.2. 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 intervals
  - stub area flag
  - interface MTU ??
  - other optional capabilities

17.3. Ospf cost

cost = 10^8 / bandwidth(bps)

Task: Description

```
(config-router)# auto-cost reference-bandwitch <bps>
```

17.4. Common OSPF protocol header format

17.4.1. Packet types

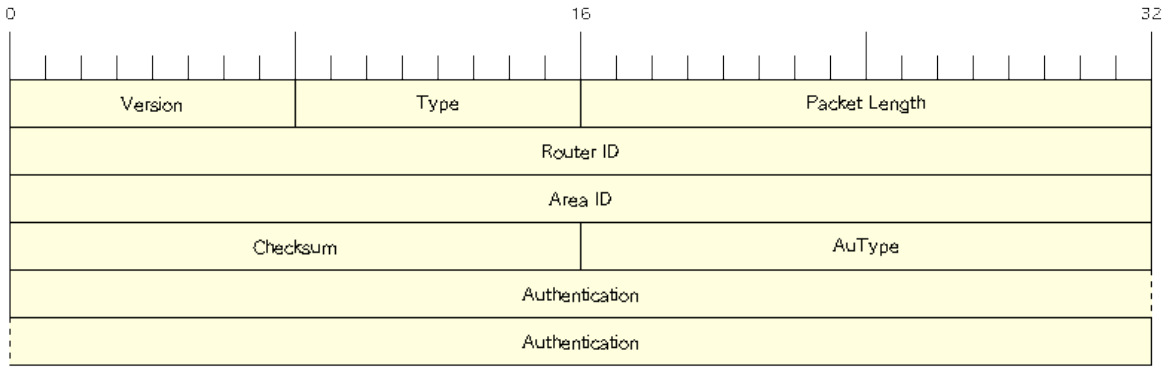


Figure 18. OSPF header format

Version

The OSPF version number (2).

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

The 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

A 64-bit field for use by the authentication scheme.

17.4.2. Hello Packet

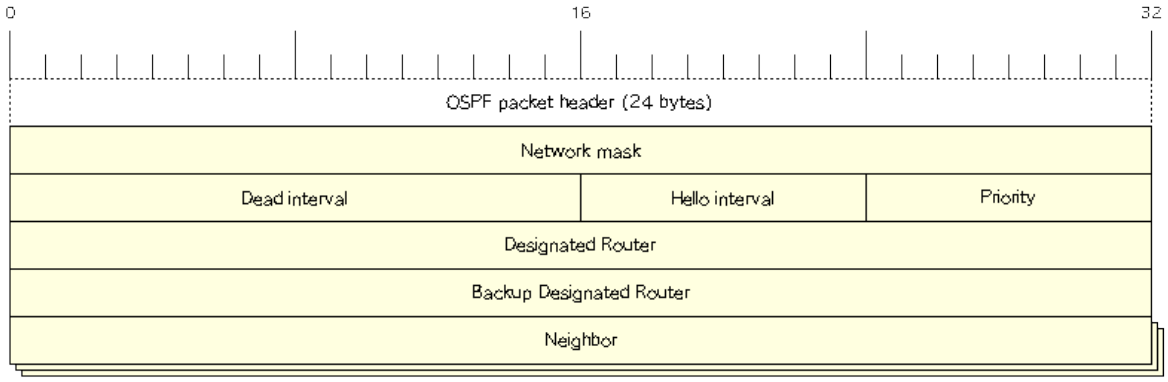


Figure 19. OSPF Hello Packet format

17.4.3. Database Description Packet

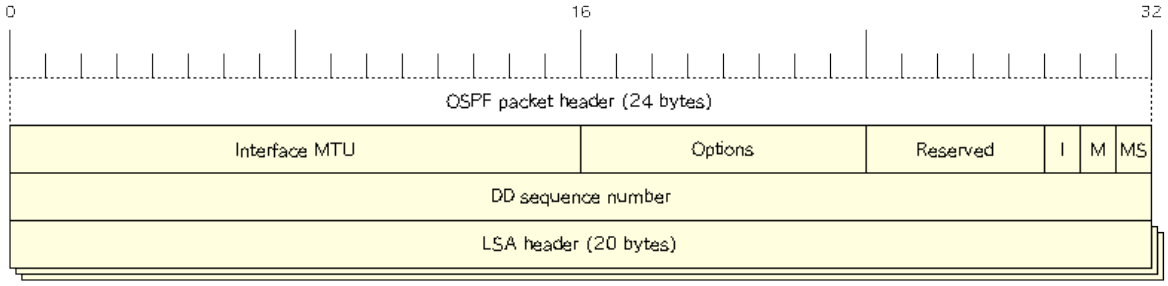


Figure 20. OSPF Hello Packet format

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

Type	Description	functionality
1	Hello	discover/maintain neighbors
2	Database description	summarize database contents
3	Link-state request	database download
4	Link-state update	databases update
5	Link-state acknowledge	flooding acknowledgement

17.4.4. Link State Request

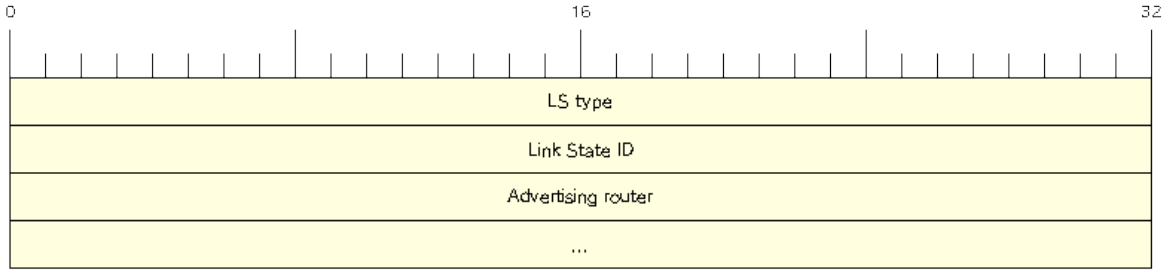


Figure 21. OSPF Link State Request format

17.4.5. Link State Update

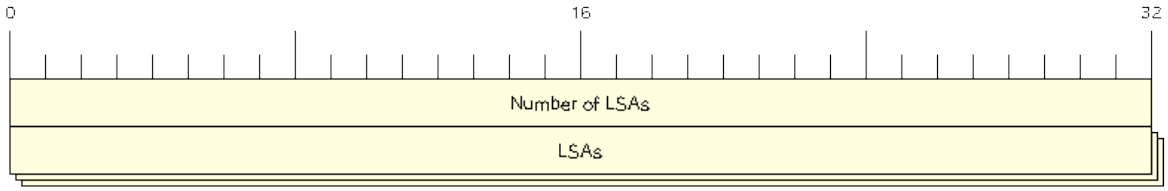


Figure 22. OSPF Link State Update format

17.4.6. Link State Acknowledgment

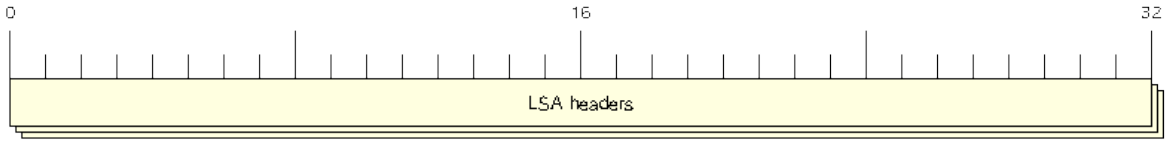


Figure 23. OSPF Link State Acknowledgment format

LSA headers

Contains LSA headers to identify the LSAs acknowledged.

17.4.7. Link-State Packets

Type 1

- Router LSA
- generated by each router for each interface in the area
- flooded only within the same area

Type 2

- Network LSA
- generated by DR
- describes the set of routers attached to a particular network
- flooded only within the area that contains the network

Type 3

- Summary inter-area LSA
- Generated by ABR
- describes inter-area routes to network

Type 4

- Summary inter-area LSA
- Generated by ABR
- describes routes to ASBR
- tells other other routers in the area how to get to the advertising router of an external route

Type 5

- AS external LSA
- originated by ASBR
- describes routes to destinations external to the AS
- flooded all over except stub areas

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.

17.5. backbone and area 0

17.6. 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
```

17.6.1. Adjacency

17.6.2. 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=0jgt

17.6.3. 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>
```

Task: Priority

```
(config-if)# ip ospf priority <0-255>
```

Task: Set the WAIT timer

```
(config-if)# ip ospf dead-timer <seconds>
```

17.6.4. network types

Point-to-point

- only 2 routers
- automatic neighbor relationships
- no DR/BDR election
- multicast hellos
- default for HDLC and PPP

broadcast

- automatic neighbor discovery
- DR/BDR election
- default for ethernet, TR, FDDI
- multicast hellos
- DR doesn’t change the next hop of advertised prefixes

Non-broadcast

- unicast hellos
- manual configuration of neighbor
- DR/BDR election
- default on Frame Relay, X.25 and SMDS

Point-to-multipoint

- multi-access, broadcast
- 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

Loopback

if Multi-Access network type then no DR/BDR election if non-broadcast, then manual configuration of neighbors

OSPF design guide: selecting interface network types

OSPF network type compatibilities

- iakfsadfj
- adsfkjasdf
- asdfjsadfj

17.6.5. 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

17.6.6. SPF throttling

17.6.7. 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

17.6.8. 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

17.6.9. 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 ofr 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.

17.7. 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>
```

Task: Prevent OSPF to advertize secondary prefixes

```
(config-if)# ip ospf <process-id> area <id> secondaries none
```

17.7.1. 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 ospg message-digest-key key-id md5 <string>
```

- Message digest

17.7.2. 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>
```

Task: configure spf throttling

```
spf ???
```

Task: Ensure that one router performs LSA translation in a NSSA area

TODO

17.8. readings

What are ospf areas and virtual links

ospf design guide: link-state advertisements

Chapter 18. BGP

Configuration guides | IP Routing | BGP

18.1. Concepts

- 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

18.1.1. 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)

18.2. 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
- Keep 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**

18.3. BGP message format

- Minimum size: 19 bytes
- Maximum size: 4096 bytes (why?)

18.3.1. KEEPALIVE

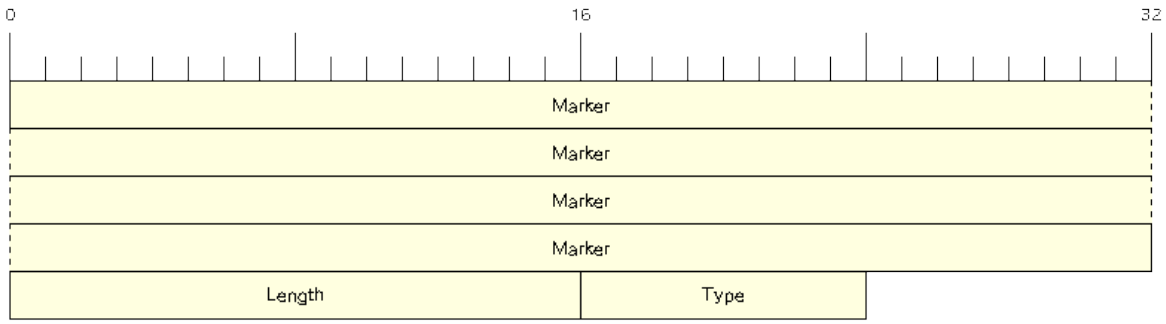


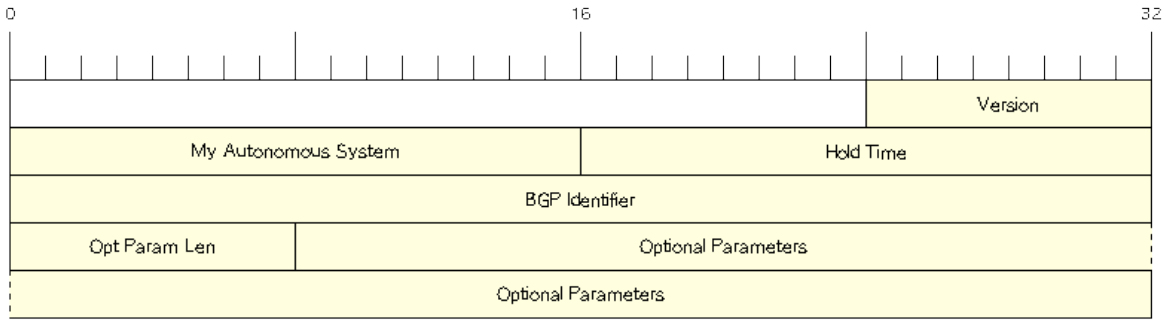
Figure 24. BGP header format

- 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)

18.3.2. 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>

18.3.3. KEEPALIVE

- Every 60 seconds
- Hold-time: 180 seconds

18.3.4. UPDATE

- Advertises a single feasible route to a peer and/or withdraws multiple unfeasible routes

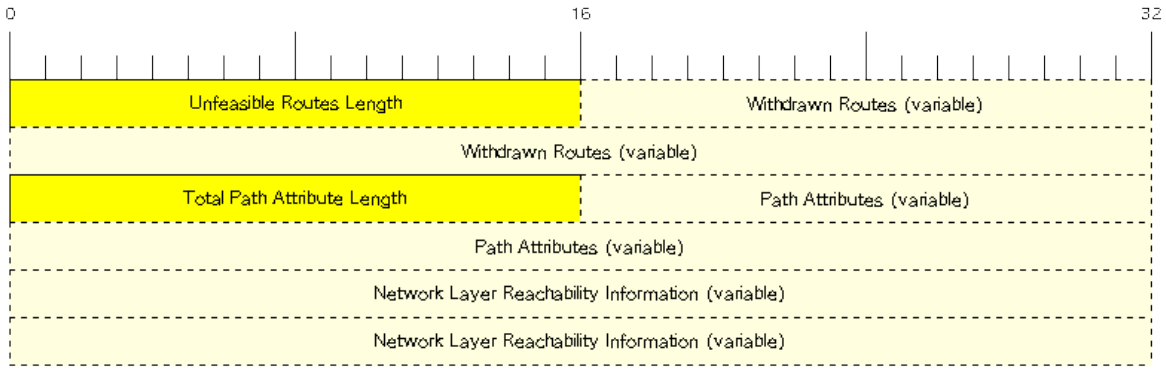


Figure 25. header format

- 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).

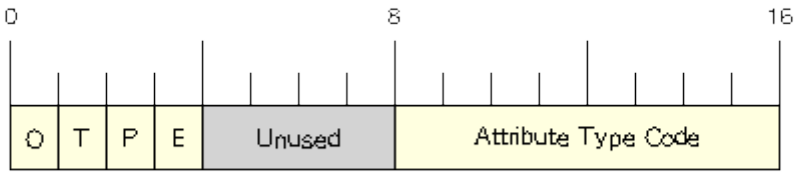


Figure 26. Attribute Type part of the Path Attributes field

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

18.3.5. NOTIFICATION

- go out in response to error, fatal condition
- torn down or reset the BGP peer session

18.3.6. BGP FSM States

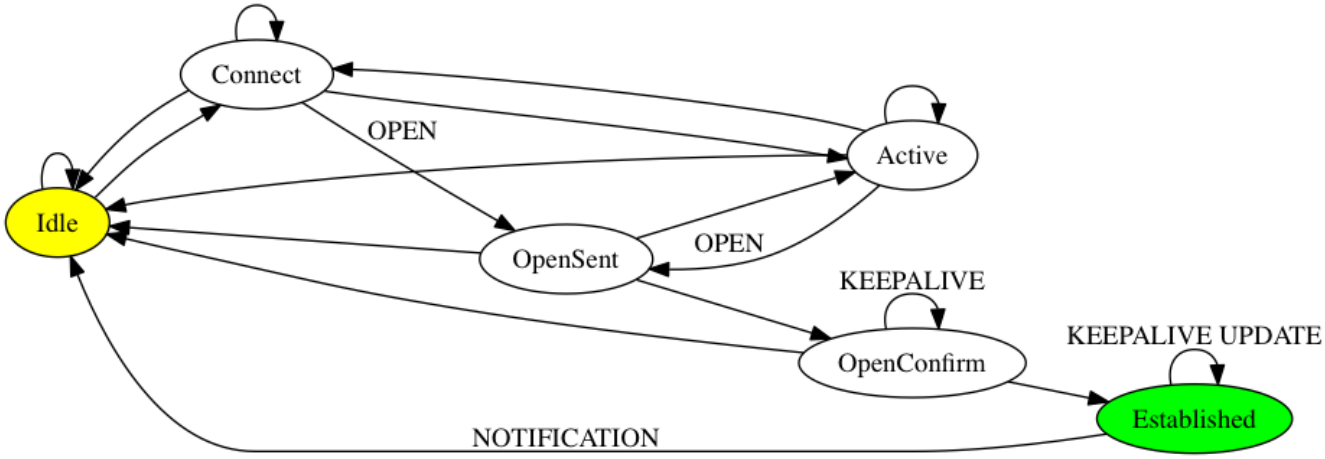


Figure 27. BGP neighbor negotiation finite state machines

- **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

18.3.7. BGP session reset

- Whenever the routing policy changes due to a configuration change
- Reset with **clear ip bgp**
- Can be hard reset, soft reset or dynamic inbound soft reset

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

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

18.3.8. 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.

18.3.9. Routing policy change management

TODO: add this part under bgp reset

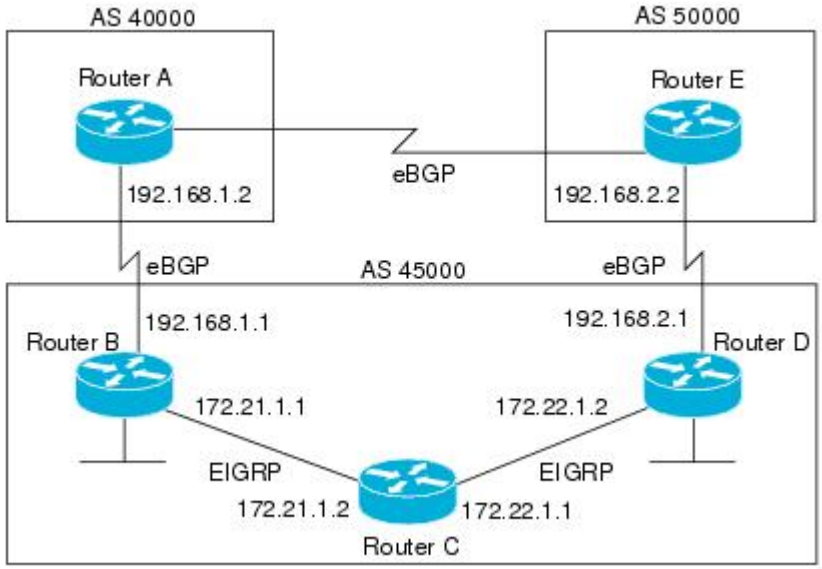
18.3.10. BGP peer groups

- 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:

18.3.11. BGP backdoor routes

- Use **network backdoor** to cause BGP to prefer EIGRP



18.3.12. Best path selection algorithm

1. highest weight
2. highest local pref
3. locally originated paths over externally originated paths
4. shortest AS path
5. lowest origin type (internal over external over incomplete)
6. lowest MED
7. eBGP paths over iBGP paths
8. lowest IGP cost
9. oldest path
10. 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)

18.3.13. 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

18.4. Configuration tasks

18.4.1. Configuring a BGP Routing Process

- Configure a bgp routing process

```
router bgp <asn>
```

- Specify a network as local to the BGP routing table

```
network <prefix> [mask  <a.b.c.d>] [route-map <name>]
```

- Configure the bgp router id

```
bgp router-id <ip-address>
```

- Set the bgp network timers

```
(config-router)# timers bgp <keepalive-seconds> <holdtime-seconds>
```

18.4.2. Configuring a BGP Peer

```
neighbor <ip-address> remote-as <asn>
```

- Specify the IPv4 address family

```
(config-router)# address-family ipv4 [unicast | multicast | vrf <name>]
```

- Enable the neighbor to exchange prefixes for the ipv4 unicast address family with the local device

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

18.4.3. Configuring a BGP Peer for the IPv4 VRF Address Family

- Associate a vpn vrf instance with an interface

```
(config-if)# interface <type> <number>
(config-if)# vrf forwarding <name>
(config-if)# ip address <prefix> <mask> [secondary [vrf <name>]]
```

- Configure a VRF routing table with the same name assigned to the VRF and enters the VRF configuration mode

```
(config)# ip vrf <name>
```

- Create routing and forwarding tables and specify the default route distinguisher for a vpn

```
(config-vrf)# rd <route-distinguisher>
```

- Create a route target extended community for a VRF

```
(config-vrf)# route-target [import | export | both] <community>
```

18.4.4. Customizing a BGP Peer

- Disable the IPv4 unicast address family for the BGP routing process

```
no bgp default ipv4-unicast
```

- Add a neighbor

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

- Add a text description with a specified neighbor

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

- Add a text description with a specified peer group

```
(config-router)# neighbor <peer-group-name> description <text>
```

- Exit address family configuration mode

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

- Disable a BGP peer or peer group

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

18.4.5. Monitoring and Maintaining Basic BGP

- Enable logging of BGP neighbor resets

```
(config-router)# bgp log-neighbor-changes
```

- 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
```

- 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.

- Apply a route map to incoming or outgoing routes

```
(config-router)# neighbor <ip-address|peer-group-name> route-map <name> [in | out]
```

18.4.6. Aggregating Route Prefixes Using BGP

- Redistribute static routes into the BGP routing table

```
(config-router)# redistribute static
```

- Create an aggregate entry in a BGP routing table

```
(config-router)# aggregate-address <prefix> <mask> [as-set]
```

- Create an aggregate route and suppress advertisements of more-specific routes to all peers

```
(config-router)# aggregate-address <prefix> <mask> [summary-only]
```

- Create an aggregate route but suppress advertisement of specified routes

```
(config-router)# aggregate-address <prefix> <mask> [suppress-map <map-name>]
```

- 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.
- Advertise selectively some BGP routes to neighbor

```
(config-router)# neighbor <ip-address> advertise-map <name-1> { exist-map <name> |
non-exist-map <name>}
```

- Inject more specific prefixes into a BGP routing table over less specific prefixes

```
(config-router)# bgp inject-map <name> exist-map <name> [copy-attributes]
```

18.4.7. Originating BGP Routes

- Advertise a default route to BGP peers

```
(config-router)# neighbor <ip-address> default-originate  [route-map <name>]
```

- Indicate a network reachable through a backdoor route

```
(config-router)# network <ip-address> backdoor
```

BGP only advertize networks in the RIB

18.4.8. Configuring a BGP Peer Group

- Create a BGP peer group

```
(config-router)# neighbor <peer-group-name> peer-group
```

- Assign a neighbor to a peer group

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

18.4.9. Modify the default output and regex match format for 4-byte ASN

```
(config-router)# bgp asnotation dot
```

18.4.10. Suppress inactive route advertisement using BGP

- Suppress inactive route advertisement

```
(config-router-af)# bgp suppress-inactive
```

18.4.11. Configure basic peer session template

- Create a peer session template

```
(config-router)# template peer-session <name>
```

- Inherit the configuration of another peer session template

```
(config-router-stmp)# inherit peer-session <template-name>
```

- 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>
```

18.4.12. configure basic peer policy template

- Create a peer policy template

```
(config-router)# template peer-policy <name>
```

- 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.

18.5. Verify

- Display the entries in the bgp routing table

```
show ip bgp [prefix] [mask]
```

- Display info about the TCP and BGP connection to neighbors

```
#  show ip bgp neigbors <ip-address>
```

- 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
```

18.6. Troubleshoot

- Verify basic network connectivity between BGP devices

```
ping vrf
```

- Clear and reset BGP neighbor sessions

```
# clear ip bgp *
```

- Clear BGP update group membership and recalcultar BGP update groups

```
# clear ip bgp update-group [ <index-group> | <ip-address> ]
```

- Display info about the processing of BGP update groups.

```
# debug ip bgp groups
```

18.7. 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

Chapter 19. 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

19.1. Administrative distance

Route source	Distance
Connected route	0
Static route	1
summary EIGRP	5
eBGP	20
internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
ODR	160
External EIGRP	170
iBGP	200
Unknown	255

19.2. Spot issues

- Loops cannot occur with one single point of redistribution
- Loops may occur with multiple points of redistribution

19.3. 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
```

19.4. Connected routes

Task: redistribute connected routes

```
redistribute connected
```

- OVerride the implicit redistribution of interfaces running the protocol X

19.5. Static routes

Task: redistribute static route

```
(config-router)# redistribute static route-map <name> metric <value>
(config-router)# default-metric <hops>
```

19.6. 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
```

19.7. 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

19.8. 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

```
```

19.9. BGP redistribution

19.9.1. IGP to BGP

- denies OSPF external routes by default

Task: redistribute OSPF into BGP

```
redistribute ospf <pid> match internal external
```

19.9.2. BGP to IGP

- iBGP routes denied by default, eBGP routes win

Part III : VPN Technologies

Chapter 20. MPLS

20.1. Concepts

- mpls vpn
  - ce : no mpls-aware
  - pe : mpls and vpn aware
  - p : no vpn aware

20.1.1. MPLS label stack

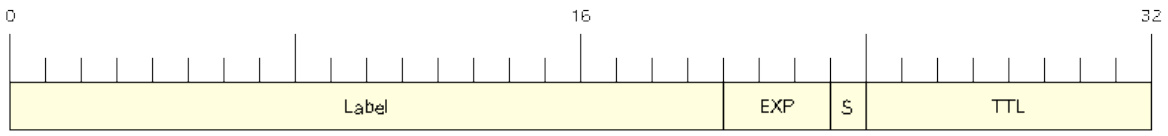


Figure 28. MPLS header format

Label
: Locally significant to the router

EXP
: Experimental, class of service

S
: Bottom-of-Stack flag

TTL
: Time to live

20.1.2. 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
```

20.1.3. 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

20.2. label distribution and control

- methods:
  - Unsolicited downstream distribution mode
  - solicited downstream distribution mode

(to be revised)

20.2.1. 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

20.3. Commands

```
show mpls forwarding-table
```

check http://www.cisco.com/en/US/docs/ios-xml/ios/mpls/command/mp-s2.html#wp4232274342

Chapter 21. GRE

21.1. Concepts

21.1.1. Tunneling

- Tunneling encapsulates data packets 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.

Components:

- **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.

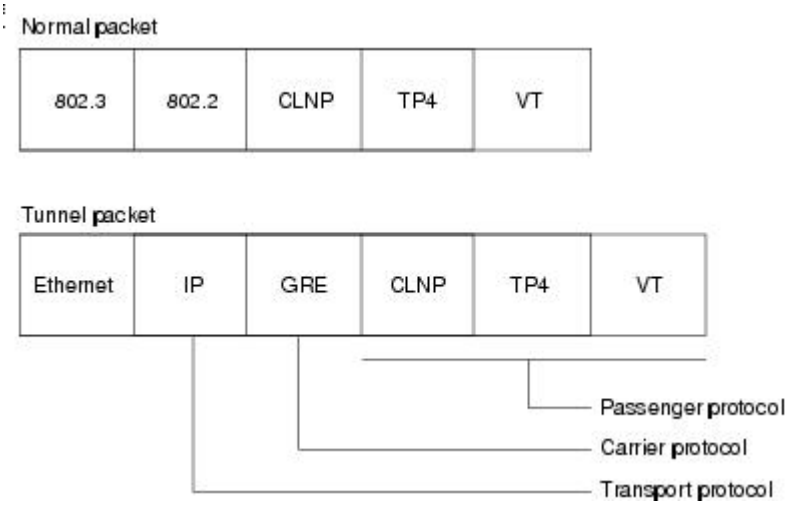
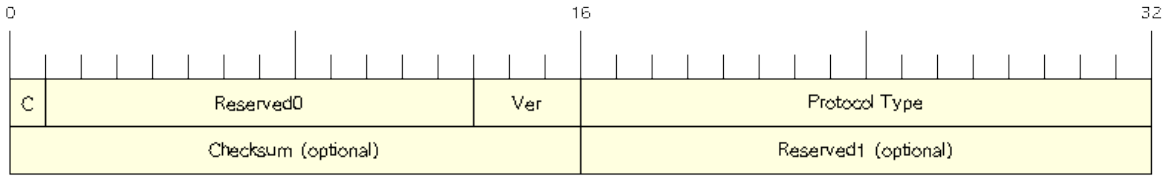


Figure 29. IP Tunneling Terminology and Concepts

21.1.2. GRE

- IP protocol 47
- RFC 2784
- RFC 2345
- RFC 1234

GRE header



Checksum present

- bit 0
- indicates that the checksum and the Reserved1 field are present

Reserved0

- 12 bits
- if any of bits 1-5 are non-zero, a receiver must discard the packet unless receiver implements RFC1701

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.

21.2. Configuration

21.2.1. Configure a 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.

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| gre multipoint}
  ip mtu bytes
  ip tcp mss mss-value
  tunnel path-mtu-discovery [age-timer {aging-mins| infinite}]
```

Create a tunnel interface

```
interface tunnel number
```

Specify a source interface for the tunnel.

The tunnel source interface can be a local physical or logical local interface, and not just an IP address

```
tunnel source {a.b.c.d | source-interface }
```

Specify the destination IP address for the tunnel

The router should have a route to this address, but not through the tunnel interface.

```
tunnel destination ip-address
```

Specify the tunnel mode

The default tunnel mode is **gre ip**.

```
tunnel mode [gre {ip | multipoint} | dvmrp | ipip | mpls | nos]
```

Adjust the GRE keepalive

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.

```
(config-if)# keepalive [ period [retries]]
```

21.2.2. 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
```

21.3. 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.

21.3.1. "%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

21.4. Questions

1. What is the minimum amount of additional header that GRE adds to a packet?
    a. 16 bytes
    b. 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

Chapter 22. DMVPN

22.1. 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

22.2. 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 23. 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
```

Chapter 24. IPSEC

Part IV : Infrastructure Security

Chapter 25. Device security

25.1. SNMP

Configuration guides | **Network Management** | **configuring SNMP support**

25.1.1. Overview

- 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

Version

v1
: original specs, weak authentication with community string

v2c
: 64-bit counters, getBulkRequest, informsRequest

v3
: authentication, encryption

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")

Packet format

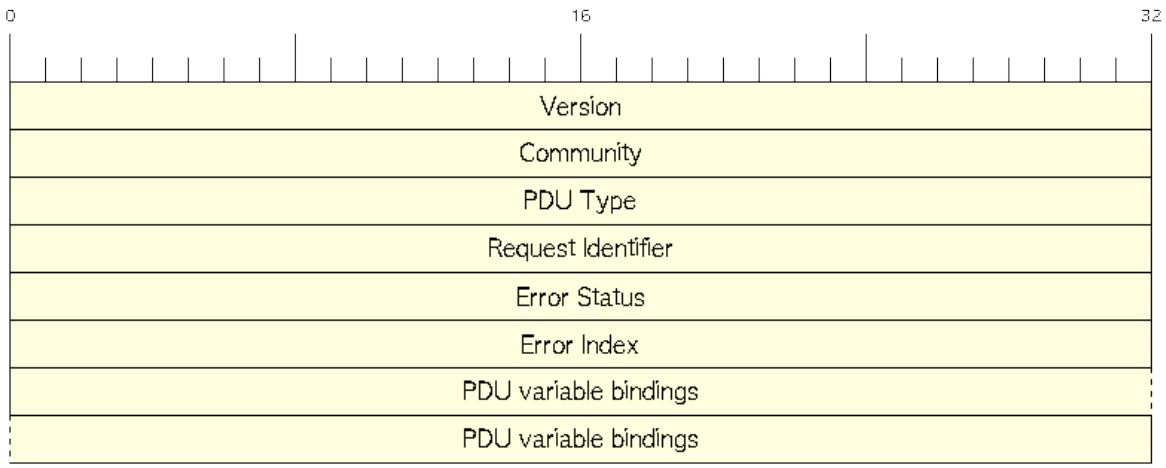


Figure 30. SNMP header format

SNMP PDU Types

1. SetRequest
2. GetRequest
3. GetNextRequest
4. GetBulkRequest
5. Trap
6. InformsRquest
7. Response

25.1.2. Configuration

Configure 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>
```

Configure SNMP v1/2

Task: Create a view

```
(config)# snmp-server view <name> <oid-tree> {included | excluded}
```

Task: Configure a community string

```
(config)# snmp-server community <string> [view <name>] [ro|rw] [<acl]
```

Task: Display community string

```
# sh snmp community
```

25.1.3. 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]
```

25.1.4. 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
```

25.1.5. Configure a device as 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
```

25.1.6. Enable the 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
```

Chapter 26. Network security

26.1. Switch security

26.2. Router security

Part V : Infrastructure Services

Chapter 27. System management

Chapter 28. QoS

Chapter 29. Network services

29.1. HSRP

Configuration Guides | First Hop Redundancy Protocols | HSRP Version 2

29.1.1. Understand

- RFC 2281
- set of routes work in concert to present the illusion of a single virtual router to the hosts on the LAN

29.1.2. Protocol

HSRP packet

MAC header | IP header | UDP packet | HSRP Packet

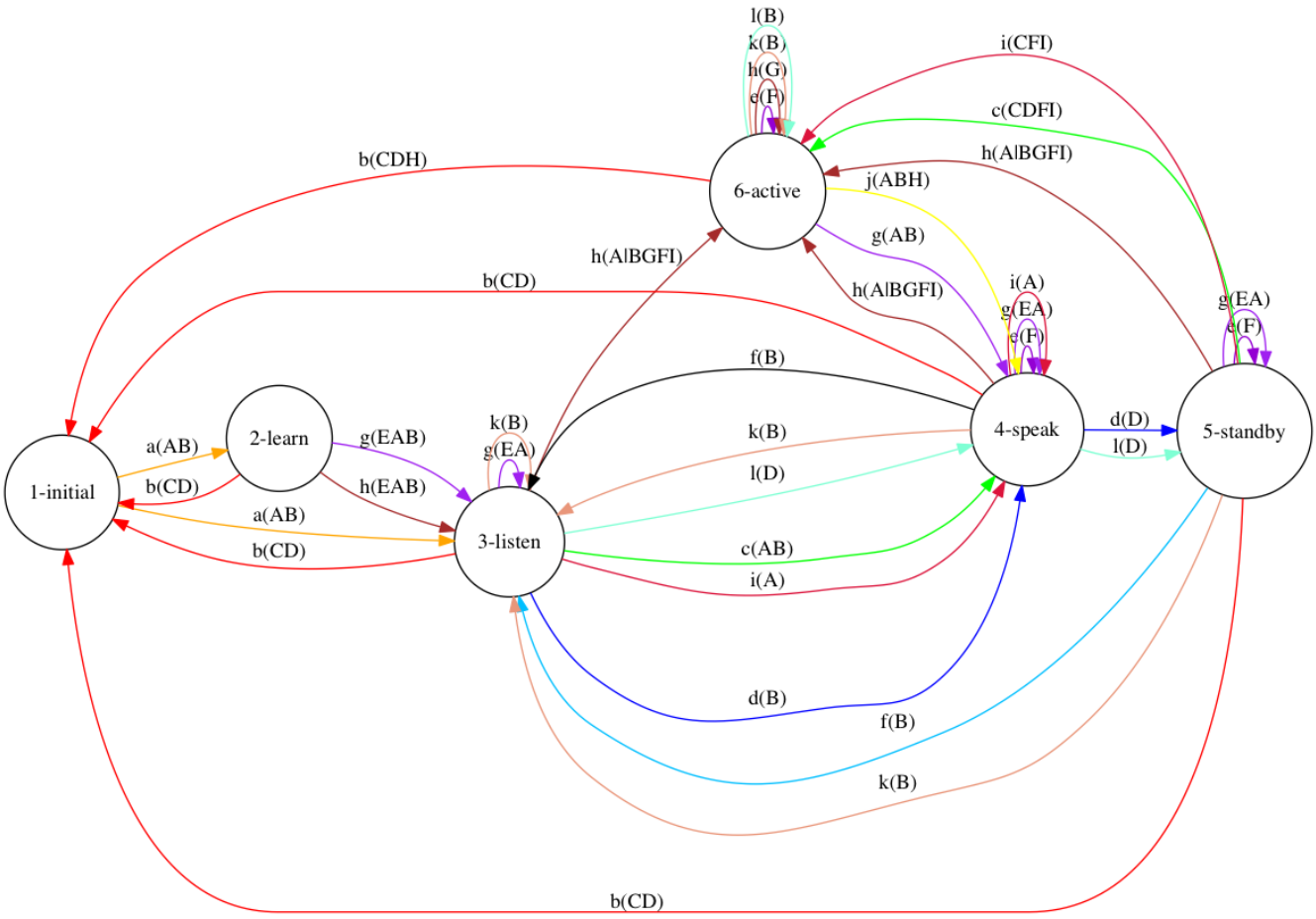


Figure 31. HSRP packet

Version

HSRP version

Opcode

- 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

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.
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.

Hellotime

- 3 seconds by default
- only meaningful in Hello messages
- configured on the router or learned from authenticated Hello message from the active router

Holdtime

- 10 seconds by default

Priority

- default value: 100
- The higher (priority || IP address) wins

Group

Authentication Data

- contains clear text password or 0x63 0x69 0x73 0x63 0x6F 0x00 0x00 0x00.

Virtual IP address

- configured or learned from active router

Finite state machine

Events

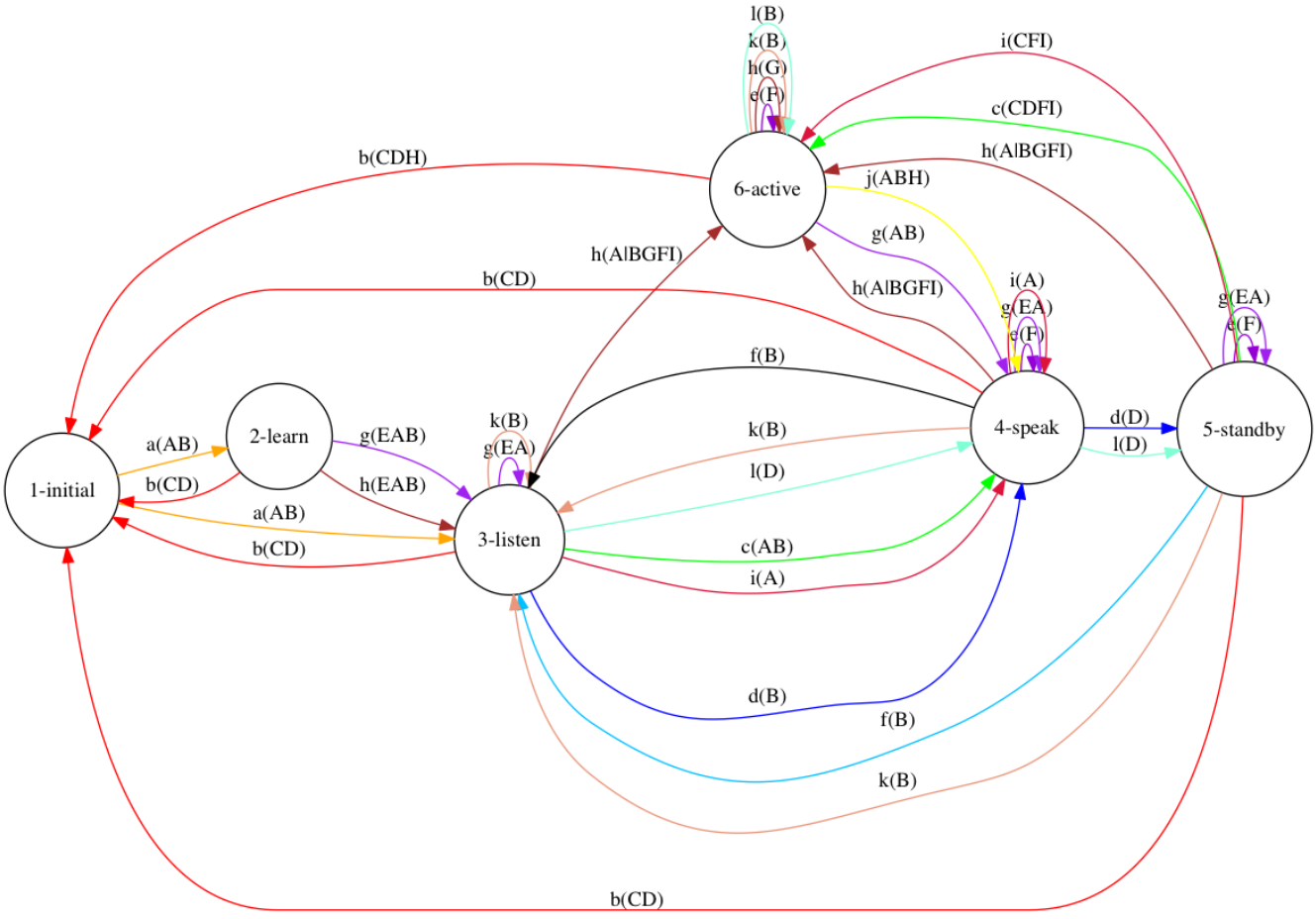
#	Description
a	HSRP is configured on an enabled interface.
b	HSRP is disabled on an interface or the interface is disabled.
c	Active timer expiry. The Active timer was set to the Holdtime when the last Hello message was seen from the active router.
d	Standby timer expiry. The Standby timer was set to the Holdtime when the last Hello message was seen from the standby router.
e	Hello timer expiry. The periodic timer for sending Hello messages has expired.
f	Receipt of a Hello message of higher priority from a router in Speak state.
g	Receipt of a Hello message of higher priority from the active router.
h	Receipt of a Hello message of lower priority from the active router.
i	Receipt of a Resign message from the active router.
j	Receipt of a Coup message from a higher priority router.
k	Receipt of a Hello message of higher priority from the standby router.
l	Receipt of a Hello message of lower priority from the standby router.

Actions

#	Action Name	Actions to be taken as part of the state machine
A	Start Active Timer	If this action occurred as the result of the receipt of a an authenticated Hello message from the active router, the Active timer is set to the Holdtime field in the Hello message. Otherwise the Active timer is set to the current Holdtime value in use by this router. The Active timer is then started.
B	Start Standby Timer	If this action occurred as the result of the receipt of an authenticated Hello message from the standby router, the Standby timer is set to the Holdtime field in the Hello message. Otherwise the Standby timer is set to the current hold time value in use by this router. The Standby timer is then started.
C	Stop Active Timer	The Active timer is stopped.
D	Stop Standby Timer	The Standby timer is stopped.
E	Learn Parameters	This action is taken when an authenticated message is received from the active router. If the virtual IP address for this group was not manually configured, the virtual IP address MAY be learned from the message. The router MAY learn Hellotime and Holdtime values from the message.

#	Action Name	Actions to be taken as part of the state machine
F	Send Hello Message	The router sends a Hello message with its current State, Hellotime and Holdtime.
G	Send Coup Message	The router sends a Coup message to inform the active router that there is a higher priority router available.
H	Send Resign Message	The router sends a Resign message to allow another router to become the active router.
I	Send Gratuitous ARP Message	The router broadcasts an ARP response packet advertising the group's virtual IP address and virtual MAC address. The packet is sent using the virtual MAC address as the source MAC address in the link layer header, as well as within the ARP packet.

Transitions



HSRP operations

- 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
- hsrp address = 0000.0c07.ACxx (where xx is the HSRP group id)

HSRP features

- 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.
- Preempt delay:
  - delay preemption for a configurate time period allowing the router to populate its routing table
  - delays starts when preemption starts in IOS > 12.0(9) otherwise when the router is reloaded.
- Interface 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)
- ICMP redirects supports
  - disable ICMP redirects on HSRP interfaces in IOS < 12.1(3)T
  - HSRP router redirects the endstation to the virtual IP address instead of a particular IP address

29.1.3. Configure

29.1.4. Verify

29.1.5. Debug

29.1.6. References

http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a0080094a91.shtml#intro

29.2. GLBP

29.2.1. Concepts

AVG active virtual gateway AVF active virtual forwarder

- AVG responds to all ARP requests with MAC from all particapating AVF

    - Not with own MAC address like HSRP
- load-balancing
    - round-robin (default)
    - weigthed
- UDP port 3222

!! GLBP configuration int f0/0.40 glbp 1 ip 172.16.26.100 glbp 1 authentication md5 key-string test

Task: Description

```
# 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
```

29.3. VRRP

Configuration Guides | First Hop Redundancy Protocols | VRRP

29.3.1. Concepts

29.3.2. Configurations

29.4. IDRP

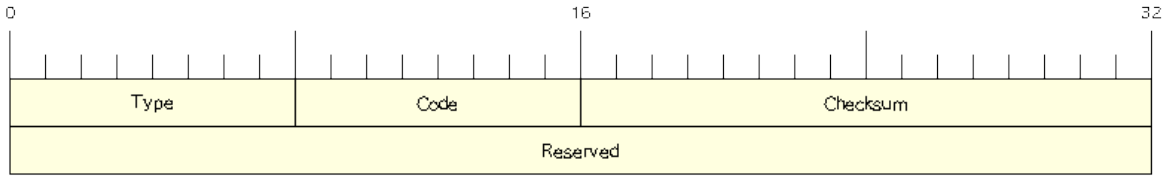
Configuration Guides | First Hop Redundancy Protocols | IRDP

29.4.1. Overview

- ICMP Router Discovery Protocol allows hosts to locate routers that can be used as gateway to reach IP-based devices on other networks.
- RFC 1256

29.4.2. Message format

ICMP Router Advertisement Message



Type

9

Code

0

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.

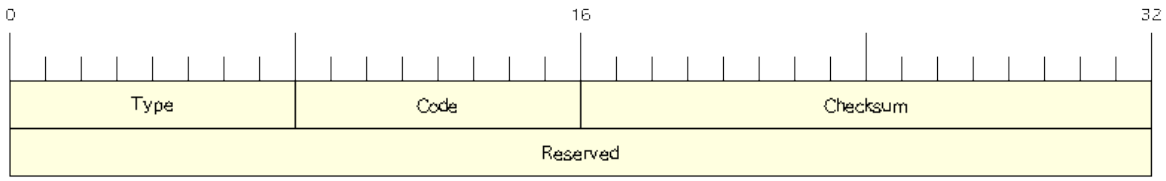
Router Address[i]

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



Type

10

Code

0

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.

29.4.3. Configuration

Task: Configure a gateway to discover routers that transmit IRDP router updates after disabling IP routing

```
no ip routing
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 prozy-advertise

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

29.5. NTP

Configuration guides | Network Management | Basic System Management | Setting Time and Calendar Services

29.5.1. Overview

- 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

29.5.2. 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>
```

29.5.3. 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.

29.5.4. 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>
```

29.5.5. 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
```

29.5.6. Authoritative server

Task:

```
(config)# ntp master
```

29.5.7. Panic treshold

Task: Reject time updates greater than the panic threshold of 1000 seconds

```
(config)# ntp panic update
```

29.5.8. 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>
```

29.5.9. external reference clock

```
# line aux <number>
# ntp refclock trimble pps none stratum <number>
```

29.5.10. 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
```

29.5.11. 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
```

29.5.12. 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
```

29.5.13. Vulnerability

- DoS for version ⇐ 4.2.4p7
- No workaround, disable NTP on the device
- Symptoms:

29.6. DHCP

- Menu:Configuration guides[IP Addressing > DHCP]

29.6.1. Overview

- Https://tools.ietf.org/html/rfc2131[RFC 2131]
  - Dynamic Host Configuration Protocol
  - Based on BOOTP
  - Client/agent relay/server model
  - UDP port 67

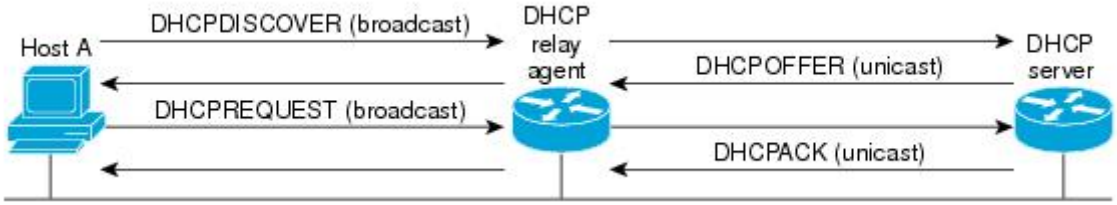


Figure 32. DHCP request for an IP address from a DHCP Server

29.6.2. Protocol operations

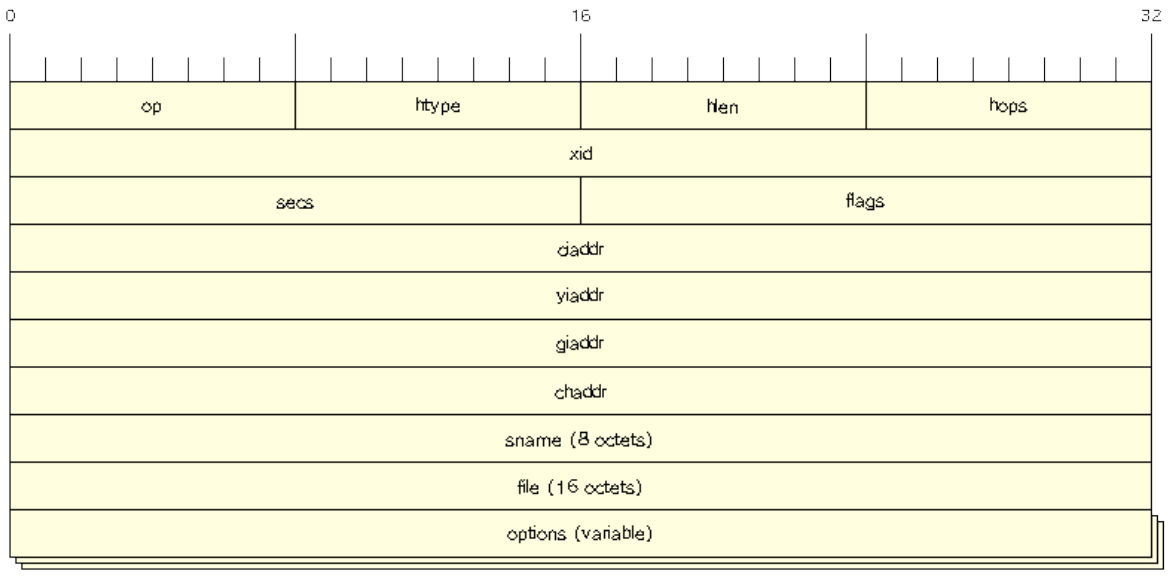


Figure 33. DHCP message

FIELD	OCTETS	DESCRIPTION
op	1	Message op code / message type. 1 = BOOTREQUEST, 2 = BOOTREPLY
htype	1	Hardware address type
hlen	1	Hardware address length

FIELD	OCTETS	DESCRIPTION
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.

29.6.3. 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
```

29.6.4. 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.

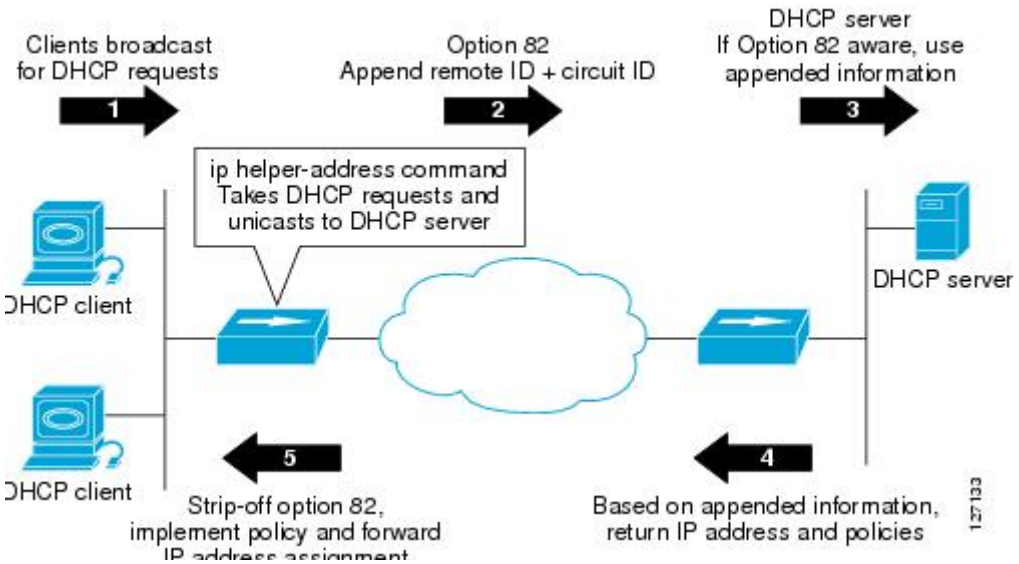
```
(config-if)# ip dhcp relay prefer known-good-server
```

- 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

29.6.5. 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
```

29.6.6. 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

29.7. NAT

29.7.1. Purpose

- NAT allows the IP network of an organization to appear from the outside to use a different IP address space than what it is actually using.
- Thus, NAT allows an organization with nonglobally routable addresses to connect to the Internet by translating those addresses into a globally routable address space.
- NAT also allows a graceful renumbering strategy for organizations that are changing service providers or voluntarily renumbering into classless interdomain routing (CIDR) blocks.
- RFC 1631.

29.7.2. 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.

29.7.3. Types of NAT

Static NAT

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

Dynamic NAT

- One local host uses an available public IP address in a pool.
- Does not conserve IP addresses.

PAT

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

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).

29.7.4. TCP load distribution for NAT

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

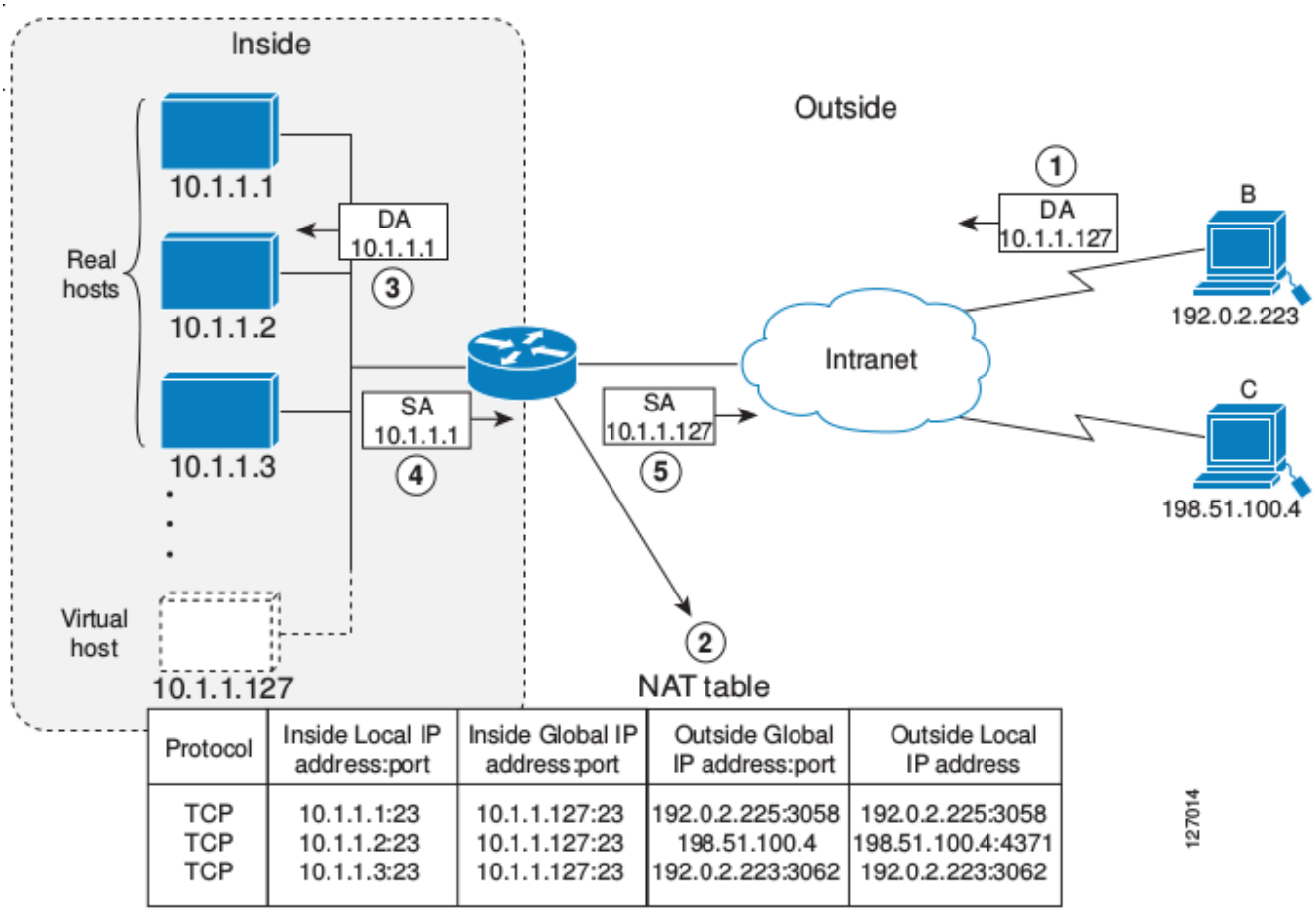


Figure 34. NAT TCP load distribution

29.7.5. NAT order of operations

Inside-to-Outside

Inside-to-Outside Outside-to-Inside

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

Configure static translation of inside source address

```
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
```

Configure dynamic translation of inside source address

```
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 inside 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
```

29.7.6. 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
```

29.7.7. 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
```

29.7.8. Configure dynamic translation of 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.

```
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
```

29.7.9. 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: Display NAT translation information

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

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
```

Chapter 30. Network optimization