Copyright information

CCIE Routing and Switching Quick Review Kit
By Krzysztof Załęski
CCIE R&S #24081, CCVP
http://www.inetcon.org
cshyshtof@gmail.com

ver. 20100507

This Booklet is NOT sponsored by, endorsed by or affiliated with Cisco Systems, Inc.

Cisco, Cisco Systems, CCIE, CCVP, CCIP, CCNP, CCNA, the Cisco Systems logo, the CCVP logo, the CCIE logo are trademarks or registered trademarks of Cisco Systems, Inc. in the United States and certain other countries.

All terms mentioned in this book, known to be trademarks or service marks belong to their appropriate right owners.

This Booklet is designed to help CCIE candidates to prepare themselves for the CCIE written and/or the lab exam. However, this is not a complete study reference. It is just a series of the author’s personal notes, written down during his pre-lab, and further studies, in a form of mind maps, based mainly on CISCO Documentation for IOS 12.4T. The main goal of this material is to provide quick and easy-to-skim method of refreshing candidate’s existing knowledge. All effort has been made to make this Booklet as precise and correct as possible, but no warranty is implied. CCIE candidates are strongly encouraged to prepare themselves using other comprehensive study materials like Cisco Documentation (www.cisco.com/web/psa/products/index.html), Cisco Press books (www.ciscopress.com), and other well-known vendor’s products, before going through this Booklet. The author of this Booklet takes no responsibility, nor liability to any person or entity with respect to loss of any information or failed tests or exams arising from the information contained in this Booklet.

This Booklet is available for free, and can be freely distributed in the form as is. Selling this Booklet in any printed or electroic form i prohibited. For the most recent version of this document, please visit http://www.inetcon.org

Did you enjoy this booklet? Was it helpful? You can share your gratitude :-) here: http://amzn.com/w/28VI9LZ9NEJF1
# Table of Contents

**Data-link technologies**
- Frame Relay 5
- PPP 6
- PPPoE 7

**Switching**
- VLAN 8
- PVST 9
- MST 10
- PortChannel 11
- L2 Port protection 11
- L2 Convergence 11
- SPAN 12
- Macro 12
- Bridging 12
- 35x0 features 12

**IP Services**
- NTP 13
- ARP 13
- CDP 13
- WCCP 13
- Routing features 14
- OER/PFR basics 15
- OER/PFR measuring 16
- OER/PFR learning 16
- OER/PFR policy 17
- OER/PFR control 17
- 1st hop redundancy 18
- NAT 19
- Management 20
- DNS 21
- DHCP 21

**Routing**
- RIPv2 22
- EIGRP part 1 23
- EIGRP part 2 24
- OSPFv2 25
- OSPF filtering 26
- OSPF neighbors 27
- OSPF LSAs 28
- BGP 29
- BGP route origin 30
- BGP aggregation 30
- BGP convergence 30
- BGP filtering 31
- BGP scalability 31
- BGP stability 31
- BGP attributes 32

**IPv6**
- IPv6 addressing 33
- IPv6 routing 34
- IPv6 tunneling 35

**Multicast**
- PIM 36
- PIM-SM 37
- PIM-DM 37
- Auto-RP 38
- Bootstrap 38
- MSDP 38
- IGMPv2 39
- Mcast features 40
- IPv6 multicast 41
- MLD 42

**Quality-of-Service**
- QoS Classify 43
- CBWFQ 44
- FIFO 44
- WRED 45
- Shaping 46
- Policing 47
- 35x0 QoS 48
- 3560 QoS 49
- 3550 QoS 50
- Compression 51
- LFI 51
- Legacy Queueing 51
- RSVP 52

**Security**
- L3 security 53
- Zone-based FW 54
- IOS IPS 55
- L2 security 56
- Other security 57

**MPLS**
- Control & Forwarding 58
- Labels 59
- MTU & TTL 59
- LDP 60
- L3 VPN 61
- PE-CE EIGRP 62
- PE-CE eBGP 62
- PE-CE Static/RIP/Other 62
- PE-CE OSPF 63
InARP triggers InARP. If LMI is disabled, InARP will not work.

**Encapsulation**

- Default FR encapsulation is CISCO
- Encapsulation frame-relay ietf
  - Frame-relay interface-dlci <#> ietf
- Frame-relay map dlci ... ietf
- Frame-relay map ip <remote-ip> <dlci> [broadcast]

**LMI (Link Management Information)**

- LMI triggers InARP. If LMI is disabled, InARP will not work.
- PIP interfaces ignore InARP messages as they only have one DLCI to them. If L2-L3 mapping is enabled, InARP will be sent to all physical interfaces.
- LMI needs only the VC to compute the DLCI to IP mapping.

**Frame-relay map ip <remote-ip> <dlci> [broadcast]**

You may also need mapping for local IP to be able to ping it. L2-L3 mapping is also required for CNR.

**Header**

- LAPF header – Link Access Procedure for Frame-Relay
  - DLCI – 10 bits (0-1023) – identifier local to each interface
  - EA – Extended address – up to 2 additional bytes of header

- InARP
  - InARP flows only across VC, it is not forwarder by routers. IP is required on intf to send InARP

- Status Enquiry: DTE->FR Switch; Status: FR Switch->DTE
  - Type-1 – keepalive (10 sec)
  - 3 misses, LMI is down

**Point-to-point**

- Frame-relay end-to-end keepalive mode (request) [bidirectional]
  - Frame-relay end-to-end keepalive event-window (recv | send) <#>
  - Frame-relay end-to-end keepalive success-events (recv | send) <#>

**Back2Back**

- Frame-relay switching
  - Frame-relay switch map ip <remote IP> <DLCI> broadcast
  - Frame-relay end-to-end keepalive mode (request) [bidirectional]
  - Frame-relay end-to-end keepalive event-window (recv | send) <#>

**Fragmentation**

- Frame-relay fragment size
  - Fragment size = delay / BW
  - Must be added on both sides, as 2 bytes fragmentation header is added
  - Fragmentation configured directly on interface with no FRTS (>12.2.13T)

**PPPofR**

- PPP over FR
  - Virtual-template is bound to DLCI. This interface is p2p then no L2-to-L3 mapping is required even if used on physical multipoint interface
  - Remote peer’s /32 IP is shown in routing table as connected (PPP behaviour)

**FR Autoinstall**

- Helper-address on staging router is required if configured router needs to upload config via TFTP. Router with TFTP server should have directed-broadcast enabled on Ethernet.
  - FR Autoinstall is enabled only on that DLCI
  - InARP is not passed through hub router, so for spokes to communicate separate static mapping is required

**Segment Routing**

- Frame-relay interface-dlci <dlci> protocol ip <ip>
  - Multilink FRFR with SNAP encapsulation

**PPPoFR**

- PPP over FR
  - Frame-relay fragment size
  - Fragment size = delay / BW
  - Must be added on both sides, as 2 bytes fragmentation header is added
  - Fragmentation configured directly on interface with no FRTS (>12.2.13T)

**Bridge**

- Frame-relay map bridge <id>... broadcast
  - Frame-relay map bridge-group <id>... broadcast
  - Static mapping is required on multipoint interfaces

**IP**

- Frame-relay interface-dlci <dlci> ip <ip>
  - IP addresses can be added on virtual-template interface
  - IP addresses can be added on virtual-template interface

**Bridging**

- Frame-relay map bridge <id>... broadcast
  - Frame-relay map bridge-group <id>... broadcast
  - Static mapping is required on multipoint interfaces

**PPP**

- Frame-relay interface-dlci <dlci> ppp virtual-template <vt-number>
  - PPP over FR

**Magic**

- Frame-relay fragment size
  - Fragment size = delay / BW
  - Must be added on both sides, as 2 bytes fragmentation header is added
  - Fragmentation configured directly on interface with no FRTS (>12.2.13T)
PAP/CHAP Authentication

One way authentication. If two-way PAP authentication is required it has to be configured the opposite way.

Client:
- hostname R1
- interface serial0/0
  - ppp pap sent-user <username> password <password>
  - ppp pap refuse [callin]

Server:
- hostname R2
- interface serial0/0
  - ppp pap send-user <username> password <password>

Two-way authentication, R2 requests R1 to auth using PAP, and R1 requests R2 to auth using CHAP.

Client:
- hostname R1
- interface serial0/0
  - ppp pap sent-user <username> password <password>
  - ppp pap refuse [callin]

Server:
- hostname R2
- interface serial0/0
  - ppp pap send-user <username> password <password>

CHAP Unidirectional 3-way challenge

Connection initiated:
- User HASH and Server HASH is compared

User HASH:
- r1801 password 1234

Server HASH:
- r3845 password 1234

Phase 1:
- User is locked up to get password

Phase 2:
- MD5
- HASH
- Random number generated by the Server, local password and ID are run through MD5 to get the HASH

Phase 3:
- User HASH and Server HASH is compared
- Server sends ACCEPT (03) or REJECT (04)
PPPoE

### Features

- **Enable on Interface**
  - `pppoes enable [group <name>]`
  - Assign PPPoE profile to an Ethernet interface. Interface will use global PPPoE profile if group is not specified.

- **protocol pppoe [group <name>]**
  - Assign PPPoE profile to VLAN subinterface. Interface will use global PPPoE profile if group is not specified.

- **interface virtual-template <number> ip unnumbered <ethernet>**
  - Verify
  - `show interfaces virtual-access <number>`
  - `clear interfaces virtual-access <number>`

### Discovery

- Host chooses one reply (based on concentrator name or on services offered). The host then sends PPPoE Active Discovery Request (PADI) packet to the concentrator that it has chosen.
- Concentrator responds with PPPoE Active Discovery Session-confirmation (PADS) packet with SESSION_ID generated. Virtual access interface is created that will negotiate PPP.

- The PPPoE Active Discovery Terminate (PADT) packet may be sent anytime after a session is established to indicate that a PPPoE session has been terminated.

### Limits

- **pppoes max-sessions <#> [threshold-sessions <#>]**
  - Specify maximum number of PPPoE sessions that will be permitted on Ethernet interface. Threshold defines when SNMP trap is sent. Max sessions depend on the platform.

- **sessions per-mac limit <per-mac-limit>**
  - Specifies the maximum number (default 100) of sessions per MAC address for each PPPoE port that uses the group.

- **sessions max limit <pppoe-session-limit> [threshold-sessions <#>]**
  - Specifies maximum number of PPPoE sessions that can be terminated on this router from all interfaces. This command can be used only in a global PPPoE profile.

### Services

- **subscriber profile <name> [refresh <min>]  pppoe service <name>**
  - Multiple services can be assigned to one profile. PPPoE server will advertise the service names to each PPPoE client that uses the configured PPPoE profile. Cached PPPoE configuration can be timed out after defined amount of time (minutes).

- **aaa new-model**
  - **aaa authorization network default group radius**

- **clear pppoe {all | interface <if> [vlan <vlan>] | rmac}**

### Verify

- **show pppoe session all**
- **show pppoe summary**

- **bba-group pppoe {<name> | global}**
  - Create BBA group to be used to establish PPPoE sessions. If global group is created it is used by all ports with PPPoE enabled where group is not specified.

- **virtual-template <number>**
  - Specifies the virtual template interface to use to clone Virtual Access Interfaces.

- **sessions per-vlan limit <per-vlan-limit>**
  - Specifies maximum number (default 100) of PPPoE sessions for each VLAN.

- **bba-group pppoe {<name> | global}**
  - Specify the dialer interface to use for cloning. A dial-on-demand keyword enables DDR functionality (idle-timeout can be configured on dialer intf). Specific service can be requested from BRAS. Service parameters are defined in RADIUS server.

- **dialer-list <dialer-group> protocol ip {permit | list <acl>}**
  - Defines which traffic brings up dialer interface.

- **interface dialer <number> encapsulation ppp  ip mtu <mtu>**
  - ! recommended 1492 for 8 byte PPPoE header

- **peer default ip address dhcp-pool <name>**
  - Assign IP address to a client from local DHCP pool.

- **pppoe client dial-pool-number <number> [dial-on-demand] [service-name <name>]**

- **interface virtual-template <number> ip unnumbered <ethernet>**
  - Assign IP address to a client from local DHCP pool.
VTP is disabled on the switch

**Transparent**

- Must be in the same domain. Default mode is Desirable on 3550 only. It is Auto on 3560.
- Messages sent every 30 sec (300sec timeout)
- If both switches support ISL and 802.1q then ISL is chosen
- Switchport mode trunk - always trunk, sends DTP to the other side
- Switchport mode access - always access, sends DTP to the other side
- Switchport mode dynamic desirable - sends negotiation DTP messages
- Switchport mode dynamic auto - Replies to negotiation DTP messages

**Server**

- Messages sent every 30 sec (300sec timeout)
- If both switches support ISL and 802.1q then ISL is chosen
- Switchport mode access - always access, sends DTP to the other side
- Switchport mode dynamic desirable - sends negotiation DTP messages
- Switchport mode dynamic auto - Replies to negotiation DTP messages

**Client**

- Messages sent every 30 sec (300sec timeout)
- If both switches support ISL and 802.1q then ISL is chosen
- Switchport mode access - always access, sends DTP to the other side
- Switchport mode dynamic desirable - sends negotiation DTP messages
- Switchport mode dynamic auto - Replies to negotiation DTP messages

**Switches must be in the same domain. Default mode is Desirable on 3550 only. It is Auto on 3560.**

**QTP**

- Always access, sends DTP to the other side

**Negotiation**

- Always access, sends DTP to the other side

**Trunking**

- Cisco proprietary protocol supporting up to 1000 VLANs
- It is a basic of device management (it includes the entire management domain)
- If no VLANs are configured (null) the first one learned is accepted, regardless of the mode (server and client).

**ISL**

- Native (non-tagged) frames received from an ISL trunk port are dropped
- Encapsulates in 26 bytes header and recalculates 4 bytes FCS trailer (real encapsulation) - total 30 bytes added to the frame

**802.1q**

- Inserts 4 byte tag after SA and recalculates original FCS. Does not tag frames on the native VLAN
- If 'only' keyword is used, the interface is mandatory (it must exist). Do not use abbreviations, full interface name must be used. However Lo1 will work, but L1 not

**VTP transparent is required**

- All hosts can be in the same subnet. VTP transparent is required

**Primary (promiscuous) VLAN**

- All devices can access this VLAN. Can send broadcast to all ports in the private VLAN (other promiscuous, trunk, isolated, and community ports)

**Community VLAN**

- All devices can access this VLAN. Can send broadcast to all ports in the private VLAN (other promiscuous, trunk, isolated, and community ports)

**Isolated VLAN**

- Can talk only to Primary. Only one can be associated with primary. Can send broadcast only to the primary ports or trunk ports

**Secondary**

- Can talk to each other and to primary. Many can be associated with primary. Can send broadcast to all primary, trunk ports, and ports in the same community VLAN

**QinQ Tunneling**

- Tagged frames (EtherType 0x8100) encapsulated within additional 4 byte 802.1q header (EtherType 0x9100), so system mtu 1504 must be added to all switches.
- Use the vlan dot1q tag native global command to configure the vlan dot1q tag native command so that all packets going out IEEE 802.1q trunk, including the native VLAN, are tagged. VLAN1 is a default native VLAN, so by default this command is required.
- The VLAN database configuration mode (if used) does not support the extended range
- Supported only in Transparent mode

**VMPS**

- The VLAN database configuration mode (if used) does not support the extended range
- Extended VLANs cannot be pruned. Supported only in Transparent mode
- Each routed port on a Catalyst 3550 switch creates an internal VLAN for its use. These internal VLANs use extended-range VLAN numbers, and the internal VLAN ID cannot be used for an extended-range VLAN. Internal VLAN IDs are in the lower part of the extended range (above vlan internal usage)

**Portfast feature is automatically enabled when voice VLAN is configured**

**Voice VLAN (3560)**

- VLAN 0
- Community VLAN
- Community VLAN 1
- Community VLAN 2
- Isolated VLAN
- Primary VLAN (3560)
- Secondary VLAN
- Promiscuous port (primary VLAN)

**Voice VLAN (3550)**

- VLAN 0
- Community VLAN
- Community VLAN 1
- Community VLAN 2
- Isolated VLAN
- Primary VLAN (3550)
- Secondary VLAN
- Promiscuous port (primary VLAN)

**Tagged frames (EtherType 0x8100) encapsulated within additional 4 byte 802.1q header (EtherType 0x9100), so system mtu 1504 must be added to all switches.**

**QinQ Tunneling**

- Tagged frames (EtherType 0x8100) encapsulated within additional 4 byte 802.1q header (EtherType 0x9100), so system mtu 1504 must be added to all switches.
- Use the vlan dot1q tag native global command to configure the vlan dot1q tag native command so that all packets going out IEEE 802.1q trunk, including the native VLAN, are tagged. VLAN1 is a default native VLAN, so by default this command is required.
- The VLAN database configuration mode (if used) does not support the extended range
- Extended VLANs cannot be pruned. Supported only in Transparent mode
- Each routed port on a Catalyst 3550 switch creates an internal VLAN for its use. These internal VLANs use extended-range VLAN numbers, and the internal VLAN ID cannot be used for an extended-range VLAN. Internal VLAN IDs are in the lower part of the extended range (above vlan internal usage)

**Portfast feature is automatically enabled when voice VLAN is configured**

**Voice VLAN (3560)**

- VLAN 0
- Community VLAN
- Community VLAN 1
- Community VLAN 2
- Isolated VLAN
- Primary VLAN (3560)
- Secondary VLAN
- Promiscuous port (primary VLAN)

**Voice VLAN (3550)**

- VLAN 0
- Community VLAN
- Community VLAN 1
- Community VLAN 2
- Isolated VLAN
- Primary VLAN (3550)
- Secondary VLAN
- Promiscuous port (primary VLAN)
PVST was supported only on ISL trunks. Bridges are not interested in local times; they use timers sent by Root Hellos.

**Blocking => Listening (15 sec) => Learning (15 sec) => Forwarding**

- **spanning-tree vlan <id> hello-time <sec>** (default is 2 sec)
- **spanning-tree vlan <id> forward-time <sec>** (default is 20 sec)
- **spanning-tree vlan <id> max-age <sec>** (default is 20 sec)

Each bridge adds 1 hello (second) to BPDUs, so each bridge shows hop count from Root. MaxAge is lowered by this value on each bridge. Max 7 hops is recommended.

Based on IEEE 802.1D standard and includes Cisco proprietary extensions such as BackboneFast, UplinkFast, and PortFast.

### timers

- **MaxAge** is lowered by this value on each bridge. Max 7 hops is recommended.
- **Bridge waits 10 Hello misses before performing STP recalculation**
- **spanning-tree vlan <id> max-age <sec>**
- **spanning-tree vlan <id> forward-time <sec>**
- **spanning-tree vlan <id> hello-time <sec>**

### Bridge states

- **Blocking** => **Listening** (15 sec) => **Learning** (15 sec) => **Forwarding**

### Port types

- **New port roles used for fast convergence**
- **point-to-point**
- **spanning-tree point-type point-to-point**

### Features

- **Backup port** - on the same switch
- **Alternate port** - on different switch

### Topology change

- **All switches need to be informed about the change to timeout CAM**
- **Switch sends TCN BPU to Root every Hello time until ACKed**
- **Upstream switch ACKs with next Hello setting Topology Change Ack (TCA) bit set**
- **Root sets TCA for next Hello BPDUs so all switches are notified about changes**
- **All switches use Forward Delay Timeout (15 sec) to time out CAM for period of MaxAge + ForwardDelay (35 sec). Root sets TCA in Hellos for that time.**

### Rapid 802.1w

- **BPDU ver 2 is used**
- **No blocking and learning state (DISCARDING, LEARNING, FORWARDING)**
- **All switches originate Hellos all the time (time capricious). Hellos are NOT relayed**
- **Neighbor querying (proposal-agreement BPDU)** is like in backbonefast, but standardized. Convergence in less than 1 sec
- **Manage only 3 Hello misses**
- **(G) spanning-tree mode rapid-pvst**

### Port roles

- **New port rules used for fast convergence**
- **point-to-point**
- **spanning-tree link-type point-to-point**

### Convergence

- **Upstream bridge sends a proposal out of DP (with proposal bit in outgoing BPDU)**
- **Downstream bridge blocks all non-designated ports and authorizes upstream bridge to put his port into forwarding state**

By Krzysztof Zaleski, CCIE #24081. This booklet is available for free and can be freely distributed in a form as is. Selling is prohibited.
### Features

**spanning-tree mode mst**

All switches within a region must have identical configuration (different configuration means different region).

VLAN-to-instance mapping is not propagated with BPDU. Only digest with region name and revision number is sent.

VLANs mapped to single MSTI must have the same topology (allowed VLANs on trunks)

When the IST converges, the root of the IST becomes the CIST regional root.

The IST and MST instances do not use the message-age and maximum-age information in the configuration BPDU to compute the STP topology. Instead, they use the path cost to the root and a hop-count mechanism (default 20).

Edge ports are designated by **spanning-tree portfast**.

Each switch decrements hop-count by 1. If switch receives BPDU with hop-count = 0, then it declares itself as a root of new IST instance.

**spanning-tree pending**

The only instance that sends and receives BPDU. All of the other STP instance information is contained in M-records, which are encapsulated within MSTP BPDU.

MST Region redistributes IST BPDU's within each VLAN to simulate PVST+ neighbor.

MSTI - Multiple Spanning Tree Instance (one or more) - RSTP instances within a region. RSTP is enabled automatically by default.

RSTP instance that extends CST inside region.

Represents MST region as CST virtual bridge to outside.

MST Region replicates IST BPDUs within each VLAN to simulate PVST+ neighbor.

CIST - (common and internal spanning tree) collection of the ISTs in each MST region, and the common spanning tree (CST) that interconnects the MST regions and single spanning trees.

CIST topology is hidden to other regions.

Each switch detects MSTIs from different region if marks the port on which it was received as boundary port.

Boundary ports exchange CIST information only. IST topology is hidden between regions.

Switch with lowest BID among all boundary switches in all regions is elected as CST root. It is also a CIST regional root within own region.

---

### MST 802.1s

**IST (MSTI 0)**

Internal Spanning Tree

**MSTI** – Multiple Spanning Tree Instance (one or more) - RSTP instances within a region. RSTP is enabled automatically by default.

### Configuration

**spanning-tree mst configuration**

name <name>
revision <number>
instance <id> vlan <range> show pending
spanning-tree mst <instance-id> root {primary | secondary}
spanning-tree mst max-hops <count>
spanning-tree mst <other STP parameters, timers>

By default, all VLANs are assigned to the IST.

Up to 16 MST instances (no limit for VLANs) – there is always one instance: 0.

The only instance that sends and receives BPDU. All of the other STP instance information is contained in M-records, which are encapsulated within MSTP BPDU.

MST Region redistributes IST BPDU's within each VLAN to simulate PVST+ neighbor.

By default, all VLANs are assigned to the IST.

External BPDUs are tunneled (CIST metrics are passed unchanged) across the region and processed only by boundary switches.

When switch detects MSTIs from different region it marks the port on which it was received as boundary port.

Boundary ports exchange CIST information only. IST topology is hidden between regions.

Switch with lowest BID among all boundary switches in all regions is elected as CST root. It is also a CIST regional root within own region.

---

### Final IST topology

IST topology is hidden to other regions.

IST topology is hidden between regions.

Switch with lowest BID among all boundary switches in all regions is elected as CST root. It is also a CIST regional root within own region.
Portfast
Immediately switches over to forwarding state. Avoid TCN generation for end hosts
BPDU guard should be enabled on that port
(G) spanning-tree portfast default
(IP) spanning-tree bpduguard default

Uplinkfast
Tracks alternate root port (second best path) to immediately switch over
(G) spanning-tree uplinkfast [max-update-rate <rate>]
If rate is 0 then no multicast flooding takes place (150 default)
BPDU guard
Supported on PVST+, rapid-PVST+, or MST
(IP) spanning-tree bpduguard default
If portfast port switches to non-portfast upon receiving BPDU,
(IP) spanning-tree bpduguard enable

Backbonefast
Indirect link failure detection, recovery within 30 sec.
If first Hello is missed switch sends Root Link Query (RLQ) out the port
where Hello was expected. If neighbor switch lost previous root too (ports
are compared for the switch and the neighbor), it informs that switch and
re-convergence (STP) occurs without waiting for MaxAge timeout (20 sec)
(G) spanning-tree backbonefast

BPDU filter
Avoids TCN generation for end hosts
BPDU guard should be enabled on that port
(G) spanning-tree portfast bpdufilter default

Etherchannel guard
A misconfiguration can occur if the switch interfaces are configured
in an EtherChannel, but the interfaces on the other device are not. If
etherchannel is not detected all bundling ports go into un-configured

(G) spanning-tree etherchannel guard misconfig

UDLD
Is not sent if is configured in Aggressive mode
Affects fiber connections only
(IP) udld {aggressive | enable}

PortChannel
802.3ad
LACP
IEEE 802.3ad
LACP
PortChannel load balancing
(R) port-channel load-balance [pr-pri | dist-mac]
XOR on rightmost bits of MAC

PortProtection
BPDU guard
Can be enabled on designated ports only. Opposite to loop guard
generates superior Hellos received on a user port (root-inconsistent)
(Ref) spanning-tree root guard
Can be enabled on non-designated ports only
(IP) spanning-tree root guard default

Loop guard
Can be enabled on non-designated ports only
(IP) spanning-tree loopguard default

Convergence
PortChannel
Cisco PAgP
(F) channel-protocol pagp
Cisco 802.1d Behaviour
on No dynamic negotiation. Forced.
off Passive
Desirable active Initiate negotiation

PortChannel disabled
Auto passive Wait for other side to initiate
Desirable active Initiate negotiation
PortChannel
802.1q
on off off
Switch with lowest system priority makes decisions about
which ports participate in bundling (switch used port-priorities)
PortChannel disabled
Auto passive Wait for other side to initiate
Desirable active Initiate negotiation

IEEE 802.3ad
LACP
(F) channel-protocol lacp
(F) channel-group <1-64> mode (passive | active)
(F) lacp port-priority <#> (default 32768, lower)
(F) lacp system-priority <#> (lower better)
show lacp sys-id

Etherchannel guard
A misconfiguration can occur if the switch interfaces are configured
in an EtherChannel, but the interfaces on the other device are not. If
etherchannel is not detected all bundling ports go into un-configured
(G) spanning-tree etherchannel guard misconfig

Etherchannel
A misconfiguration can occur if the switch interfaces are configured
in an EtherChannel, but the interfaces on the other device are not. If
etherchannel is not detected all bundling ports go into un-configured
(G) spanning-tree etherchannel guard misconfig

Cisco PAgP
(F) channel-protocol pagp
(Aggressive mode attempts to reconnect once a second 8 times before err-disabling
Uses L2 probes every 15 sec to mac 01:00:0C:CC:CC:CC. Must be ACKed by remote end.
If configured for the first time it is not enabled untill first Hello is received
(IP) pagp port-priority <#>
The physical port with the highest priority (default is 128) that is operational and has
membership in the same EtherChannel is the one selected for PAgP transmission

Etherchannel
A misconfiguration can occur if the switch interfaces are configured
in an EtherChannel, but the interfaces on the other device are not. If
etherchannel is not detected all bundling ports go into un-configured
(G) spanning-tree etherchannel guard misconfig

IEEE 802.3ad
LACP
PortChannel
802.3ad
LACP
PortChannel load balancing
(R) port-channel load-balance [pr-pri | dist-mac]
XOR on rightmost bits of MAC

PortProtection
BPDU guard
Can be enabled on designated ports only. Opposite to loop guard
generates superior Hellos received on a user port (root-inconsistent)
(Ref) spanning-tree root guard
Can be enabled on non-designated ports only
(IP) spanning-tree root guard default

Loop guard
Can be enabled on non-designated ports only
(IP) spanning-tree loopguard default

Convergence
PortChannel
Cisco PAgP
(F) channel-protocol pagp
Cisco 802.1d Behaviour
on No dynamic negotiation. Forced.
off Passive
Desirable active Initiate negotiation

PortChannel disabled
Auto passive Wait for other side to initiate
Desirable active Initiate negotiation
PortChannel
802.1q
on off off
Switch with lowest system priority makes decisions about
which ports participate in bundling (switch used port-priorities)
PortChannel disabled
Auto passive Wait for other side to initiate
Desirable active Initiate negotiation

IEEE 802.3ad
LACP
(F) channel-protocol lacp
(F) channel-group <1-64> mode (passive | active)
(F) lacp port-priority <#> (default 32768, lower)
(F) lacp system-priority <#> (lower better)
show lacp sys-id
You cannot monitor outgoing traffic on multiple ports. Only 2 SPAN sessions per switch.

You can monitor incoming traffic on a series or range of ports and VLANs.

Span (Rx) SPAN – catch frames before any modification or processing is performed by the switch. Destination port still receives a copy of the packet even if the actual incoming packet is dropped by ACL or QoS drop.

You can monitor incoming traffic on a series or range of ports and VLANs.

Span (Rx) SPAN – catch frames before any modification or processing is performed by the switch. Destination port still receives a copy of the packet even if the actual incoming packet is dropped by ACL or QoS drop.

You must create the RSPAN VLAN in all switches that will participate in RSPAN (VTP can be used). The reflector port (Cat 3550 only) loops back untagged traffic to the switch. It is invisible to all VLANs. The traffic is then placed on the RSPAN VLAN and flooded to any trunk ports that carry the RSPAN VLAN. No access port must be configured in the RSPAN VLAN; it cannot be 1 or 1002-1005.

All routable traffic received on a bridged interface is routed to other routable interfaces as if it is coming directly from BVI.

The MAC address-table move update feature allows the switch to provide rapid bidirectional convergence when a primary link goes down and the standby link begins forwarding traffic.

Flex Links are a pair of a Layer 2 interfaces where one interface is configured to act as a backup to the other. Users can disable STP and still retain basic link redundancy. Preemption can be enabled so traffic goes back to primary link after it comes back up. Backup link does not have to be the same type. STP is automatically disabled on Flex Link ports. The MAC address-table move update feature allows the switch to provide rapid bidirectional convergence when a primary link goes down and the standby link begins forwarding traffic. The MAC address-table move update feature allows the switch to provide rapid bidirectional convergence when a primary link goes down and the standby link begins forwarding traffic.

FlexLink

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

FlexLink

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link

Flex Link
Control messages – reading and writing internal NTP variables

Request/Update messages – actual time synchronization

Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet use Subnetwork Access Protocol (SNAP).

Request/Update messages – actual time synchronization

Features

(IF) arp timeout <sec>

Proxy ARP

(IF) arp timeout <sec>

Features

(IF) no ip proxy-arp

Secure ARP

(IF) inet host-cache

Authentication

(IP) inet host-cache

Authentication

(IF) inet host-cache

Authentication

(IF) inet host-cache

Authentication

(IF) inet host-cache

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication

Authentication
### Network classes

<table>
<thead>
<tr>
<th>Network classes</th>
<th>Protocol #</th>
<th>Administrative Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 126</td>
<td>A</td>
<td>Directly connected</td>
</tr>
<tr>
<td>127</td>
<td>Loopback</td>
<td>0</td>
</tr>
<tr>
<td>128 – 191</td>
<td>B</td>
<td>Static to interface</td>
</tr>
<tr>
<td>192 – 239</td>
<td>C</td>
<td>Static to INR</td>
</tr>
<tr>
<td>234 – 255</td>
<td>D</td>
<td>EIGRP Summary</td>
</tr>
<tr>
<td>240 – 255</td>
<td>Reserved</td>
<td>EIGRP Internal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OSPF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EIGRP External</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BGP local</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unknown (not valid)</td>
</tr>
</tbody>
</table>

### Port numbers

<table>
<thead>
<tr>
<th>Port numbers</th>
<th>7kg/udp</th>
<th>9kg/udp</th>
<th>13/udp</th>
<th>19/udp</th>
<th>67/udp</th>
<th>68/udp</th>
<th>113/udp</th>
<th>123/udp</th>
<th>137/udp</th>
<th>138/udp</th>
<th>139/udp</th>
<th>161/udp</th>
<th>162/udp</th>
<th>176/udp</th>
<th>514/udp</th>
<th>514/tcp</th>
<th>520/udp</th>
<th>521/udp</th>
</tr>
</thead>
</table>

### Protocol #

<table>
<thead>
<tr>
<th>Protocol #</th>
<th>Administrative Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICMP</td>
<td>Directly connected</td>
</tr>
<tr>
<td>IGMP</td>
<td>0</td>
</tr>
<tr>
<td>IP</td>
<td>1</td>
</tr>
<tr>
<td>TCP</td>
<td>6</td>
</tr>
<tr>
<td>UDP</td>
<td>17</td>
</tr>
<tr>
<td>RSVP</td>
<td>47</td>
</tr>
<tr>
<td>IPv6</td>
<td>46</td>
</tr>
<tr>
<td>GRE</td>
<td>47</td>
</tr>
<tr>
<td>OSPF</td>
<td>47</td>
</tr>
<tr>
<td>BGP</td>
<td>200</td>
</tr>
</tbody>
</table>

### Port numbers

<table>
<thead>
<tr>
<th>Port numbers</th>
<th>7kg/udp</th>
<th>9kg/udp</th>
<th>13/udp</th>
<th>19/udp</th>
<th>67/udp</th>
<th>68/udp</th>
<th>113/udp</th>
<th>123/udp</th>
<th>137/udp</th>
<th>138/udp</th>
<th>139/udp</th>
<th>161/udp</th>
<th>162/udp</th>
<th>176/udp</th>
<th>514/udp</th>
<th>514/tcp</th>
<th>520/udp</th>
<th>521/udp</th>
</tr>
</thead>
</table>

### Port numbers

<table>
<thead>
<tr>
<th>Port numbers</th>
<th>7kg/udp</th>
<th>9kg/udp</th>
<th>13/udp</th>
<th>19/udp</th>
<th>67/udp</th>
<th>68/udp</th>
<th>113/udp</th>
<th>123/udp</th>
<th>137/udp</th>
<th>138/udp</th>
<th>139/udp</th>
<th>161/udp</th>
<th>162/udp</th>
<th>176/udp</th>
<th>514/udp</th>
<th>514/tcp</th>
<th>520/udp</th>
<th>521/udp</th>
</tr>
</thead>
</table>

### Redistributed protocol

<table>
<thead>
<tr>
<th>Protocol #</th>
<th>Administrative Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPF</td>
<td>110</td>
</tr>
<tr>
<td>RIP</td>
<td>140</td>
</tr>
<tr>
<td>EIGRP</td>
<td>160</td>
</tr>
<tr>
<td>IS-IS</td>
<td>160</td>
</tr>
<tr>
<td>BGP</td>
<td>200</td>
</tr>
</tbody>
</table>

### Port numbers

| Port numbers | 7kg/udp | 9kg/udp | 13/udp | 19/udp | 67/udp | 68/udp | 113/udp | 123/udp | 137/udp | 138/udp | 139/udp | 161/udp | 162/udp | 176/udp | 514/udp | 514/tcp | 520/udp | 521/udp |
|--------------|---------|---------|--------|--------|--------|--------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|

### Redistribution

1. Track remote router with RTR: track rtr 1 reachability delay down <sec> up <sec>
2. Create bogus static routing. Routing to tracked RTR. Although the route is pointed to null0, which is always available, the route will be in the routing table only if status of tracked resource is UP:
   - route protocol redistributed from <protocol> to <protocol>
3. Create prefix-list covering bogus route and assign it to route-map (conditional default route injection)
4. Originate a default route (RIP in this example) only if route-map result is true, meaning the remote router is reachable:
   - route-map TST permit 1.1.1.1 redistribute protocol <protocol>

### Reliability routing

- Can be used to track next-hop if it's not directly connected
- Tracking two or more events with boolean expression
- Track 3 list boolean and object

### Advanced Object Tracking

- Hub router can automatically discover stub networks while the stub routers still use a default route to the hub (also learned via ODR): *protocol 0.0.0.0 [160/1] via *
- ODR can convey on any network the interface of the last hop.
- It discovers information about stub networks but does not provide any routing information to the stub routers. Information is conveyed by a CDP (Cisco Discovery Protocol).

### Protocol number 47

- By default configured tunnel does not have the ability to bring down the line protocol of either tunnel endpoint, if the far end is unreachable. If keepalive is enabled, NAT cannot be used for GRE packets.

### Distribution

- If AD is manipulated, and two protocols have the same AD, the tie-breaker is the default, original AD for each protocol.
- distance (default) <ip> <mask> <seq> refers to the advertising router ad - which routes get new distance

### Match Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Prefix-list A permit 0.0.0.0 ge 8 le 32 le 32.0.0.0 match ip</th>
<th>Prefix-list B permit 128.0.0.0 ge 4 ge 16 le 32 le 255.255.255.255 match ip</th>
<th>Prefix-list C permit 192.0.0.0 ge 12 le 32 le 31.255.255.255 match ip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lo0</td>
<td>10.0.0.0.1</td>
<td>20.0.0.0.2</td>
<td>30.0.0.0.2</td>
</tr>
<tr>
<td>GRE Proto</td>
<td>S: 10.0.0.1</td>
<td>S: 20.0.0.2</td>
<td>S: 30.0.0.2</td>
</tr>
<tr>
<td>IP</td>
<td>D: 10.0.0.1</td>
<td>D: 20.0.0.2</td>
<td>D: 30.0.0.2</td>
</tr>
</tbody>
</table>

### Step 1

1. Track remote router with RTR: track rtr 1 reachability delay down <sec> up <sec>
2. Create bogos static routing. Routing to tracked RTR. Although the route is pointed to null0, which is always available, the route will be in the routing table only if status of tracked resource is UP:
   - route protocol redistributed from <protocol> to <protocol>
3. Create prefix-list covering bogos route and assign it to route-map (conditional default route injection)
4. Originate a default route (RIP in this example) only if route-map result is true, meaning the remote router is reachable:
   - route-map TST permit 1.1.1.1 redistribute protocol <protocol>

### Step 2

1. Track remote router with RTR: track rtr 1 reachability delay down <sec> up <sec>
2. Create bogos static routing. Routing to tracked RTR. Although the route is pointed to null0, which is always available, the route will be in the routing table only if status of tracked resource is UP:
   - route protocol redistributed from <protocol> to <protocol>
3. Create prefix-list covering bogos route and assign it to route-map (conditional default route injection)
4. Originate a default route (RIP in this example) only if route-map result is true, meaning the remote router is reachable:
   - route-map TST permit 1.1.1.1 redistribute protocol <protocol>
The list of traffic class entries is called a Monitored Traffic Class (MTC) list. The entries in the MTC list can be profiled either by automatically learning the traffic or by manually configuring the traffic classes (both methods can be used at the same time).

**Features**
- **Learn (BR)**: BR learns traffic based on the default prefix and sends it to MC.
- **Apply Policy (MC)**: Every rule has three attributes: scope (traffic class), action (insert a policy), and per external link (overwrite the previous).
- **Enforce (BR)**: Routing can be manipulated with artificialy injected more-specific routes. Measured prefixes’ parent route must be reachable by the BR. The limits on this operation are not explicitly specified.
- **Verify (MC)**: After the policies are introduced, OER will verify that the optimized traffic is flowing through the preferred exit or entry links at the network edge.

**Phases**
- **Wheel**
  - **Define Interfaces on BR**: OER compares the results with a set of configured low and high thresholds for each metric and policy. If the results are not within the configured thresholds, OER initiates route changes when one of the following occurs: traffic class goes out-of-profile, exit link becomes unavailable, or the exit link becomes available.
  - **Define Interfaces on MC**: MC sends the policy changes to the BRs.
  - **Define Interfaces on BR**: BRs apply the policy changes to the network.

**Basics**
- **Interface <if> [external | internal]**: Define interfaces which are used on BR (must exist on BR).
- **logging**: Enables logging for a master controller (notice level).
- **keepalive <sec>** (Defaults to 60 sec.): Keepalive between MC and BR. Default is 60 sec.
- **config**: Log settings.

**Interface virtual-template 1**
- **ip nat inside source list 1 Interface virtual-template 1 orv overlay oer NAT awareness for SOHO. NAT session will remain in case of route change via second ISP.

**Border Router**
- **Border <ip> [key-chain <name>]**
- **local <intf>**
- **master <ip> [key-chain <name>]**: Identifies source for communication with an OER MC.

**Master Controller**
- **oer master**: Enables OER master controller.
- **config**: Enable OER master controller.

**OER/PfR**
- **Enables syslog messages for a master controller**
- **Define Interfaces on BR**: OER-managed exit links to forward traffic. At least two for OER-managed domain, at least one on each BR.
- **Internal interfaces – used for communication between MCC and BRs. Loopback interface should be configured if MC and BR are on the same router. Configured only on BR.
- **Authentication**: OER-based authentication for SOHO. NAT session will remain in case of route change via second ISP.
- **Keepalive between MC and BR**
- **Keepalive between BRs**

**OER/PfR**
- **Learn (BR)**
- **Apply Policy (MC)**
- **Enforce (BR)**
- **Verify (MC)**

**Features**
- **Learn (BR)**: BR learns traffic based on the default prefix and sends it to MC.
- **Apply Policy (MC)**: Every rule has three attributes: scope (traffic class), action (insert a policy), and per external link (overwrite the previous).
- **Enforce (BR)**: Routing can be manipulated with artificialy injected more-specific routes. Measured prefixes’ parent route must be reachable by the BR. The limits on this operation are not explicitly specified.
- **Verify (MC)**: After the policies are introduced, OER will verify that the optimized traffic is flowing through the preferred exit or entry links at the network edge.

**Phases**
- **Wheel**
  - **Define Interfaces on BR**: OER compares the results with a set of configured low and high thresholds for each metric and policy. If the results are not within the configured thresholds, OER initiates route changes when one of the following occurs: traffic class goes out-of-profile, exit link becomes unavailable, or the exit link becomes available.
  - **Define Interfaces on MC**: MC sends the policy changes to the BRs.
  - **Define Interfaces on BR**: BRs apply the policy changes to the network.

**Basics**
- **Interface <if> [external | internal]**: Define interfaces which are used on BR (must exist on BR).
- **logging**: Enables logging for a master controller (notice level).
- **keepalive <sec>** (Defaults to 60 sec.): Keepalive between MC and BR. Default is 60 sec.
- **config**: Log settings.

**Interface virtual-template 1**
- **ip nat inside source list 1 Interface virtual-template 1 orv overlay oer NAT awareness for SOHO. NAT session will remain in case of route change via second ISP.

**Border Router**
- **Border <ip> [key-chain <name>]**
- **local <intf>**
- **master <ip> [key-chain <name>]**: Identifies source for communication with an OER MC.

**Master Controller**
- **oer master**: Enables OER master controller.
- **config**: Enable OER master controller.

**OER/PfR**
- **Enables syslog messages for a master controller**
- **Define Interfaces on BR**: OER-managed exit links to forward traffic. At least two for OER-managed domain, at least one on each BR.
- **Internal interfaces – used for communication between MCC and BRs. Loopback interface should be configured if MC and BR are on the same router. Configured only on BR.
- **Authentication**: OER-based authentication for SOHO. NAT session will remain in case of route change via second ISP.
- **Keepalive between MC and BR**
- **Keepalive between BRs**

**OER/PfR**
- **Learn (BR)**
- **Apply Policy (MC)**
- **Enforce (BR)**
- **Verify (MC)**

**Features**
- **Learn (BR)**: BR learns traffic based on the default prefix and sends it to MC.
- **Apply Policy (MC)**: Every rule has three attributes: scope (traffic class), action (insert a policy), and per external link (overwrite the previous).
- **Enforce (BR)**: Routing can be manipulated with artificialy injected more-specific routes. Measured prefixes’ parent route must be reachable by the BR. The limits on this operation are not explicitly specified.
- **Verify (MC)**: After the policies are introduced, OER will verify that the optimized traffic is flowing through the preferred exit or entry links at the network edge.
Delay, Jitter, MOS are monitored using IP SLA probes. Loss – counters are incremented if retransmission takes place (repeated sequence number in TCP segment).

Reachability – tracks SYN without corresponding ACK. Throughput – total number of packets sent (all types of traffic).

Delay – only for TCP flows (RTT between sending TCP segment and receipt of ACK). Learned probes (ICMP) are automatically generated when a traffic class is learned using the NetFlow.

To test the reachability of the specified target, OER performs a route lookup in the BGP or static routing tables for the specified target and external interface.

A probe target is assigned to traffic class with the longest matching prefix in MTC list.

Reachability – tracks SYN without corresponding ACK. Passive mode monitors both Active and Passive – both methods enabled together (different than fast failover). Default mode.

After external interface is configured for BR, OER automatically monitors utilization of that link. BR reports link utilization to MC every 20 sec.

Default frequency is 60 sec. Fast failover - all exits are continuously probed using active monitoring and passive monitoring. Probe frequency can be set to a lower frequency than for other monitoring modes, to allow a faster failover capability. Failover within 3 sec.

By default active probes are sourced from an OER managed external interfaces.

Mixed modes

Active Probe

Passive probe

OE/PfR

Mixed modes

Passive

Active

Throughput

Set maximum utilization range for all OER-managed exit links. OER keeps the links within utilization range, relative to each other. Ensures that the traffic load is distributed. If the range falls below threshold OER will attempt to move some traffic to use the other exit link to even the traffic load.

Automatic learning (learn)

Manual learning

OER/PfR

Learning

Automatic learning based on a protocol or port number (application learning). Aggregate only flows matching specified criteria. There can be multiple protocol entries for automatic application learning.

Prefix-list ge 32 is used to specify only inclusive prefix. Only named extended ACLs are supported.

OER will not control inside prefixes unless there is exact match in BGP.

Prefix-list ge is not used and & 32 is used to specify only inclusive prefix.

Mixed modes

Active and Passive – both methods enabled together (different than fast failover). Default mode.

Forced target assignment

OER border address source interface ip

By default active probes are learnt from an OER managed external interfaces.

Show oer master active-probes [apair | forced]
The RMON engine on a router polls the SNMP MIB variables locally, no need to waste resources on SNMP queries. When the value of the MIB variable crosses a raising threshold RMON creates a log entry and sends an SNMP trap. No more events are generated for that threshold until the opposite falling threshold is crossed.

```
rmon alarm <number> <MIB OID> <interval> {delta | absolute} rising-threshold <value> [event-number] falling-threshold <value> [event-number] [owner <string>]
rmon event <number> [log] [trap <community>] [description <string>] [owner <string>]
```

**Logging**

```
logging facility <facility-type>
```

**Accounting**

```
accounting (IF) ip accounting access-violation
Access-violation requires ACL to be applied on the interface. It cannot me a named ACL.
appling accounting-threshold <threshold>
The default value is 512 source/destination pairs. This default results in a maximum of 12,028 bytes of memory usage for each of the five databases, active and check pointed.
```

**Netflow**

```
(IP) ip route-cache flow
show ip flow-export (IP) ip flow-aggregation cache (autonomous_system | destination-prefix | prefix | protocol-port | source-prefix)
```

**Management**

```
show ip flow-top-talkers top <#> sort by (packets | bytes)
rmon alarm (IF) ip route-cache flow
show ip flow-aggregation cache (autonomous_system | destination-prefix | prefix | protocol-port | source-prefix)
```

**CPU threshold**

```
smp-server enable traps cpu threshold
Enables CPU thresholding violation notification as traps and inform requests
smp-server host <ip> traps <community> cpu
Sends CPU traps to the specified address
```

**Synchronization**

```
logging synchronous
Refresh existing config line if log message overwrites it
logging buffered <size> <level>
logging rate-limit console all <msg/sec>
```

**Syslog**

```
logging host <ip> [transport {udp | tcp} port <port>]
logging trap <severity>
```

**Services**

```
exception core-file <name>
```

**TCLSH**

```
foreach VAR {10.0.0.1 10.0.0.2} puts [exec "ping $VAR"]
```

**Archiving**

```
archive log config
show archive log ...
archive path ... write-memory
archive config differences <config1> <config2>
show archive config differences <config1> <config2> Displays differences in DIFF style
show archive config incremental-diffs <config>
```

**Core dump**

```
exception core-file (name)
```

**Misc**

```
except protocol ftp
```

**Core dump**

```
show ip cache flow
```

**Dampening**

```
dampening [half-life] <reuse> <suppress> <max> [restart]
```

**Syslog**

```
busy-message <hostname> <message>
```

**Interface**

```
interface (IF) ip accounting mac-address {input | output}
interface (IF) ip accounting output-packets
interface (IF) ip accounting precedence {input | output}
interface (IF) ip accounting-threshold <threshold>
The default value is 512 source/destination pairs. This default results in a maximum of 12,028 bytes of memory usage for each of the five databases, active and check pointed.
interface (IF) ip route-cache flow
show ip flow-export (IP) ip flow-aggregation cache (autonomous_system | destination-prefix | prefix | protocol-port | source-prefix)
```

**Archiving**

```
archive log config
show archive log ...
archive path ... write-memory
archive config differences <config1> <config2>
show archive config differences <config1> <config2> Displays differences in DIFF style
show archive config incremental-diffs <config>
```

**Core dump**

```
except protocol ftp
```

**Misc**

```
except protocol ftp
```

**Core dump**

```
show ip cache flow
```

**Dampening**

```
dampening [half-life] <reuse> <suppress> <max> [restart]
```

**Syslog**

```
busy-message <hostname> <message>
```

**Interface**

```
interface (IF) ip accounting mac-address {input | output}
interface (IF) ip accounting output-packets
interface (IF) ip accounting precedence {input | output}
interface (IF) ip accounting-threshold <threshold>
The default value is 512 source/destination pairs. This default results in a maximum of 12,028 bytes of memory usage for each of the five databases, active and check pointed.
interface (IF) ip route-cache flow
show ip flow-export (IP) ip flow-aggregation cache (autonomous_system | destination-prefix | prefix | protocol-port | source-prefix)
```

**Archiving**

```
archive log config
show archive log ...
archive path ... write-memory
archive config differences <config1> <config2>
show archive config differences <config1> <config2> Displays differences in DIFF style
show archive config incremental-diffs <config>
```

**Core dump**

```
except protocol ftp
```

**Misc**

```
except protocol ftp
```

**Core dump**

```
show ip cache flow
```
By default RIP sends only RIPv1 messages but listens to both RIPv1 and RIPv2. If either version 1 or version 2 is manually defined, only this version is sent and received on all interfaces, regardless of per-interface configuration.

**Neighbors**

Unicast updates to specified peer. Used in conjunction with passive-interface on broadcast interface, as the above command does not suppress sending multicast updates, and peer will receive double updates.

**Interface**

If a neighbor is set as a unicast neighbor, all unicast updates are sent to that neighbor. If the neighbor is set as a broadcast neighbor, all updates are sent to the network as a broadcast.

**No-Neighbor Mode**

If no-neighbor mode is enabled, RIP will not send any updates to the network. This is useful when you want to prevent RIP updates from being sent to a specific network.

**No-Summary Mode**

If no-summary mode is enabled, RIP will not generate any summary routes. This is useful when you want to prevent RIP from summarizing routes.

**Default Route**

Default route is a route that is used by routers when no other route is available. It is a route that a router uses to forward packets to a default gateway.

**Summary**

Summary is a route that is used by routers when no other route is available. It is a route that a router uses to forward packets to a default gateway.

**Filtering**

Filtering is a process used to control the flow of data across a network. It is used to prevent certain types of traffic from being forwarded or to forward certain types of traffic to specific locations.

**Auto-Summarization**

Auto-summarization is a feature that allows a router to aggregate multiple subnets into a single summary route. This is useful when you have a large number of subnets and you want to reduce the amount of routing information that needs to be sent.

**Auto-Summarization**

Auto-summarization does not override summary-address only if split-horizon is not enabled and summary-address and interface IP share the same major network. If enabled on an interface and the interface is enabled or is a multicast interface, summary-address is advertised on that interface. If disabled on an interface, summary-address is not advertised on that interface.

**Split-Horizon**

Split-horizon is a feature that prevents routers from advertising routes that they have learned from other routers.

**Default Route**

Default route is a route that is used by routers when no other route is available. It is a route that a router uses to forward packets to a default gateway.

**Passive Interface**

Passive interface is a feature that allows a router to prevent updates from being received by a specified interface. This is useful when you want to prevent RIP updates from being received by a specific interface.

**Interface**

If an interface is marked as passive, RIP will not accept any updates from that interface. This is useful when you want to prevent RIP updates from being received by a specific interface.
**EIGRP Part 1**

Protocol 88 multicasted to 224.0.0.10
3 tables: neighbor, topology, routing
8 packets based on TLV. Hello, Update, Ack, Query, Goodbye, SIA Query, SIA Reply

**Components**
- Reliable Transport Protocol (RTP)
- Neighbor Discovery/Recovery
- Diffusing Update Algorithm (DUAL)

**Features**
- RTP

**Neighbor Discovery/Recovery**
- No ip split-horizon eigrp <as>
- Split horizon enabled for all interfaces except physical with FR
- EIGRP traffic uses max 50% of bandwidth for control traffic (not data)

**Timers**
- Hello and Holdtime are announced but do not have to match. Router usespeer’s values
- Hello and Hold must be changed together, not like in OSPF where Hello changes Holdtime

**Authentication Per-interface MD5 only**
- (IF) ip authentication mode eigrp <as> md5
- (IF) ip authentication key-chain eigrp <as> <key-name>

**Security**
- Key rotation with accept-lifetime and send-lifetime options in key-chain

**Default Route**
- Router uses own interface bandwidth if it’s less than advertised by peer
- Internal paths are preferred over external paths regardless of metric
- Internal paths are preferred over external paths regardless of metric
- Delay = 10 microsec. Delay is cumulative
- Offset-list can be used to manipulate inbound and outbound metric (delay is changed with offset-list !!!)
- Some suppressed routes can be still advertised with leak-map, which has to be used only if summarization is applied on physical interface (not available on subinterfaces at all). For subinterfaces PPP can be used to create VirtualTemplate physical interface.
- More specific prefix can be also leaked with more specific summary route. Both leak-map and more specific summary can co-exist together.
- If Null0 route is poisoned with distance 255, the null0 route is not installed in local routing table, but the summary is still advertised on that interface.

**Summarization**
- No auto-summary

**Auto summarization is enabled by default.
- (IF) ip summary-address eigrp <as> <network> <mask> <distance>
- Default AD for EIGRP summary is 5. Route is pointed to Null0
- If Null0 route is poisoned with distance 255, the null0 route is not installed in local routing table, but the summary is still advertised on that interface.

**Metric**
- Multi-hop router is required to have the same metric assigned to it via EIGRP on all interfaces.
- (IF) ip summary-address eigrp <as> <network> <mask> <distance>

**Part 1**

Null0 is an interface, so 0.0.0.0 will be treated as connected network and announced via EIGRP

If network is received by one router as candidate-default [*100.1.0.0], and you do not want to propagate this network as default use no default-information allowed out. This network will be passed forward, but not as default candidate anymore

Default metric weights:
- TOD = 0 (languages): K1 (BW) + K2 (BW) + K3 (Distance) * K5 (Reliability) + K4
- Default Metric = 256 * K1 + (BW / 10) + Delay / 10

Sample composite metric calculation for default K-values:
- BW: 10,000,000 / 100Mb = 100 Delay: (5000 loopback + 100 Ethernet) / 10ms = 510 Metric: (100 + 510) * 256 = 156160

**Neighbor**<ip> <intf>
- Send hellos as unicast, and suppress sending any hellos via 224.0.0.10 on specified interface. Static configuration is required on all other peers on the same interface too.

**Hello (default) not acknowledged**
- Must be in the same AS and K-values must match

- Source of Hello is primary subnet on interface

- Passive-interface <ip> <intf>
- Stop hellos on specified interface

**Timers**
- Hello and Holdtime are announced but do not have to match. Router usespeer’s values
- Hello and Hold must be changed together, not like in OSPF where Hello changes Holdtime
- No response to query is received within this time, the route is declared SIA
- Multicast Flow Timer – if no ACK is received from peer the update is retransmited individually

**Authentication Per-interface MD5 only**
- (IF) ip authentication mode eigrp <as> md5
- (IF) ip authentication key-chain eigrp <as> <key-name>

**Security**
- Key rotation with accept-lifetime and send-lifetime options in key-chain

**Default Route**
- Router uses own interface bandwidth if it’s less than advertised by peer
- Internal paths are preferred over external paths regardless of metric
- Delay T = 10 microsec. Delay is cumulative
- Offset-list can be used to manipulate inbound and outbound metric (delay is changed with offset-list !!!)

**Summarization**
- No auto-summary

**Auto summarization is enabled by default.
- (IF) ip summary-address eigrp <as> <network> <mask> <distance>
- Default AD for EIGRP summary is 5. Route is pointed to Null0
- If Null0 route is poisoned with distance 255, the null0 route is not installed in local routing table, but the summary is still advertised on that interface.

**Metric**
- Multi-hop router is required to have the same metric assigned to it via EIGRP on all interfaces.
- (IF) ip summary-address eigrp <as> <network> <mask> <distance>

**Part 1**

Null0 is an interface, so 0.0.0.0 will be treated as connected network and announced via EIGRP

If network is received by one router as candidate-default [*100.1.0.0], and you do not want to propagate this network as default use no default-information allowed out. This network will be passed forward, but not as default candidate anymore

Default metric weights:
- TOD = 0 (languages): K1 (BW) + K2 (BW) + K3 (Distance) * K5 (Reliability) + K4
- Default Metric = 256 * K1 + (BW / 10) + Delay / 10

Sample composite metric calculation for default K-values:
- BW: 10,000,000 / 100Mb = 100 Delay: (5000 loopback + 100 Ethernet) / 10ms = 510 Metric: (100 + 510) * 256 = 156160

**Neighbor**<ip> <intf>
- Send hellos as unicast, and suppress sending any hellos via 224.0.0.10 on specified interface. Static configuration is required on all other peers on the same interface too.

**Hello (default) not acknowledged**
- Must be in the same AS and K-values must match

- Source of Hello is primary subnet on interface

- Passive-interface <ip> <intf>
- Stop hellos on specified interface

**Timers**
- Hello and Holdtime are announced but do not have to match. Router usespeer’s values
- Hello and Hold must be changed together, not like in OSPF where Hello changes Holdtime
- No response to query is received within this time, the route is declared SIA
- Multicast Flow Timer – if no ACK is received from peer the update is retransmited individually

**Authentication Per-interface MD5 only**
- (IF) ip authentication mode eigrp <as> md5
- (IF) ip authentication key-chain eigrp <as> <key-name>

**Security**
- Key rotation with accept-lifetime and send-lifetime options in key-chain

**Default Route**
- Router uses own interface bandwidth if it’s less than advertised by peer
- Internal paths are preferred over external paths regardless of metric
- Delay T = 10 microsec. Delay is cumulative
- Offset-list can be used to manipulate inbound and outbound metric (delay is changed with offset-list !!!)

**Summarization**
- No auto-summary

**Auto summarization is enabled by default.
- (IF) ip summary-address eigrp <as> <network> <mask> <distance>
- Default AD for EIGRP summary is 5. Route is pointed to Null0
- If Null0 route is poisoned with distance 255, the null0 route is not installed in local routing table, but the summary is still advertised on that interface.

**Metric**
- Multi-hop router is required to have the same metric assigned to it via EIGRP on all interfaces.
- (IF) ip summary-address eigrp <as> <network> <mask> <distance>
By default EIGRP will load balance across 4 equal paths.

**maximum-paths <1..16>**

By default EIGRP will load balance across 4 equal paths.

**traffic-share min**

If more paths exist than allowed choose the ones over different physical interfaces.

**traffic-share balanced**

- less packets to lower-bandwidth paths (default)

EIGRP automatically redistributes IGRP routes if the IGRP process is in the same autonomous system.

**Load balancing**

No default metric, must be manual set when redistributing into EIGRP

Metric is derived automatically for routes redistributed from connected, static or other EIGRP processes.

**default-metric <bw> <delay> <reliability> <loRD> <mtu>**

Variance 2 in the below example means that any route with FD < 20 (2 * 10) will be used to load-balance traffic in appropriate ratio proportional to the metric.

**Load balancing – loss patters to lower-bandwidth paths (default)**

**Routing Information Protocol (EIGRP)**

Part 2

**Redistribution and filtering**

**Tags**

Tags can be added to routes to manipulate route entries and mutual redistribution

**IP EIGRP automatically redistributes IGRP routes if the IGRP process is in the same autonomous system.**

Traffic-share min

If more paths exist than allowed choose the ones over different physical interfaces.

**traffic-share balanced**

- less packets to lower-bandwidth paths (default)

Redistribution

- less packets to lower-bandwidth paths (default)

**Distance**

Distance set for specific prefix originated by specific source (works ONLY for internal routes, external are not matched at all)

**Successor**

Successor – feasible successor that is currently being used as the next hop to the destination

FD – feasible distance – best distance to remote network (successor route) installed in routing table

FS – feasible successor – not a successor route, but still meets feasibility condition (RD < FD)

**FD**

FD – feasible distance – best distance to remote network (successor route) installed in routing table

**FS**

FS – feasible successor – not a successor route, but still meets feasibility condition (RD < FD)

**Leak-map**

Leak-map can be used to redistribute additional routes (even those learned from other peers, regardless of stub route types to be advertised), but querying is still suppressed, as it is a stub.

Leaked routes can be limited per-neighbor by specifying interface route-map LEAK permit 10 match ip address <acl> match interface <if>

**Query scoping**

Query scoping is used to avoid SIA and to minimize convergence

- if some route fails

**Route summarization**

- if peer does not have queried prefix but it has summarized route it instantly replies negatively without doing own query

**Routing Information Protocol (EIGRP)**

Part 2

**Topology (DUAL)**

**Stub**

Stub router

**Route summarization**

- if peer does not have queried prefix but it has summarized route it instantly replies negatively without doing own query

**Route summarization – if peer does not have queried prefix but it has summarized route it instantly replies negatively without doing own query**

**Route summarization**

- if peer does not have queried prefix but it has summarized route it instantly replies negatively without doing own query

**Route summarization**

- if peer does not have queried prefix but it has summarized route it instantly replies negatively without doing own query

**Route summarization**

- if peer does not have queried prefix but it has summarized route it instantly replies negatively without doing own query
Hello: 10 sec LAN, 30 sec NBMA; Dead: 4x Hello (40 sec LAN, 120 sec NBMA) – counts down

LSA Refresh: 30 min - Each router originating LSA re-floods it with incremented Seq every 30 min (Link State Refresh interval)

LSA Age: 60 min - Each router expects LSA to be refreshed within 60 min

1 sec Dead with 250 ms Hello (Fast Hello Feature)

Poll interval: on NBMA Hello to neighbor marked down – 60 sec

Delay for re-sent LSUs (if not ACKed) default 5 sec

Hello: 10 sec LAN, 30 sec NBMA; Dead: 4x Hello (40 sec LAN, 120 sec NBMA) – counts down

Set cost for a default route automatically generated by an ABR. Useful if many ABRs exist. By default cost of default is 1

Suppose LSA3 default which is generated automatically

Area <id> default-cost <cost>

DR and BDR election. Hello sent as unicast (30 / 120)

If regular router originates default it becomes ABR. If ABR originates default it is not an ABR

Default

OSPF does not support summary-address 0.0.0.0 to generate a default

DR and BDR election. Hello sent as multicast (10 / 40)

Area <id> default-cost <cost>

Default is compared only if routes are of the same type

Area <id> nssa no-summary

Point-to-multipoint

Default autocost reference: 100,000,000/BW bps

Default cost for a default route automatically generated by an ABR. Useful if many ABRs exist. By default cost of default is 1

Point-to-multipoint

DR and BDR election. Hello sent as broadcast (10 / 40)

Point-to-multipoint nonbroadcast

Cost

DHCP

P-to-P

Non-broadcast

IP protocol 89; 224.0.0.5 All OSPF Routers; 224.0.0.6 All DR Routers

Totaly stubby

Area <id> default-cost <cost>

Area <id> nssa default-cost <cost>

Router-ID can be any dotted-decimal number (0.0.0.1), not necessarily valid IP

Hello: 10 sec LAN, 40 sec NBMA; Dead: 4x Hello (40 sec LAN, 120 sec NBMA)

The segment is seen as collection of /32 endpoints (regardless of netmask), not a transit subnet

Point-to-point

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint

Point-to-multipoint nonbroadcast

Point-to-multipoint nonbroadcast

Point-to-multipoint
OSPF does not support summary-address to supernet 0.0.0.0 to generate a default. When OSPF is enabled on an interface, it always advertises directly connected subnets.

External routes can be summarized only on ASBR which redistributes those routes. Cost is taken from the smallest cost of the component routes.

Suppress all prefixes except loopbacks and passive interfaces (OSPF) prefix-suppression.

Inter-area (LSA1 and LSA2) routes can be summarized on ABR. Component route area <id> range <prefix> <mask> [cost <cost>] must exist in area <id>. Cost of summary is the lowest cost of more specific prefixes.

Suppress all prefixes on an interface (loopbacks and passive too). Takes precedence over router-mode command. (IF) ip ospf prefix-suppression [disable].

Summarization since 12.1 summary will automatically create null0 route to prevent loops. It can be disabled no discard-route {internal | external}.

Area <id> nssa translate type7 suppress-fa. If "subnets" keyword is omitted, router redistributes classful subnets, not additional summary can be created for that more specific route (multiple summaries). OSPF default metric (E2) of redistributed IGP routes = 20 (subnets) and 1 for BGP.

Secondary subnets on an interface covered by the network router ospf <process> network <net> <wildcard> area <id>. Redistribution and command are advertised as Stub route origin and network matches primary intf, OSPF is enabled also on unnumbered (hellos are sent) redistribute max-prefix <max routes> <% warning> [warning-only] [ignore-time <min>] [ignore-count <#>] [reset-time <min>] protection.

Only internal, non-self-originated routes are counted. When the keyword is used, the OSPF process never enters the ignore state. When max is reached the process goes into Ignore-state for ignore-time (5 min). If going into ignore-mode repeats ignore-count (5 times) times the process is down forever. If process stays stable for reset-time (10 min) minutes the ignore-count timer is reset to 0. The clear ip ospf process does not clear this counter.

Filters ("in" means into routing table) ANY routes which LSADB chooses to add into routing table. Can be used on ANY router, as it affects only local router's routing table (even if route-map is used). The router will not be used as transit, unless it is the only path. The only exception to "in" is when prefix being filtered is coming from area 0, then prefix will be filtered from routing table AND a database protection.

Filtering configured on ABRs. VL can stay active after authentication is applied as it is an on-demand circuit (hellos suppressed). Virtual-Link VL has no IP address, so it does not carry data traffic, only control-plane. VL is an interface in area 0 (must be authenticated if area 0 is authenticated).

OSPF stub router advertises max metric for all routes, which are not originated by that router. Local routes are advertised with normal metric. OSPF stub router advertises max metric for all routes, which are not originated by that router. Local routes are advertised with normal metric. The router will not be used as transit, unless it is the only path. The only exception to "in" is when prefix being filtered is coming from area 0, then prefix will be filtered from routing table AND a database protection.

Filtering configured on ABRs. VL can stay active after authentication is applied as it is an on-demand circuit (hellos suppressed). Virtual-Link VL has no IP address, so it does not carry data traffic, only control-plane. VL is an interface in area 0 (must be authenticated if area 0 is authenticated).

OSPF stub router advertises max metric for all routes, which are not originated by that router. Local routes are advertised with normal metric. OSPF stub router advertises max metric for all routes, which are not originated by that router. Local routes are advertised with normal metric. The router will not be used as transit, unless it is the only path. The only exception to "in" is when prefix being filtered is coming from area 0, then prefix will be filtered from routing table AND a database protection.
**OSPF Neighbors**

**Neighbor**

- Hello
- 224.0.0.5 MAC:0100.5E00.0005
- Sourced from interface primary subnet

**Adjacency**

- Adjacency is possible on unnumbered interfaces with different subnets but if they are in the same area.
- Primary interface must be covered by network statement, not an `ip ospf` interface command which is not inherited.
- Interface statements overlap, most specific are used first.

**States**

- To successfully form an adjacency parameters must match: Authentication, Area, DR/BDR capability, Timers.

**Attempt**

- Applies only to manually configured neighbors on NBMA networks.
- A router sends packets to a neighbor in all PollInterval instead of HelloInterval.
- Link Hello has been seen from the neighbor in the last HelloDeadInterval.

**2-Way**

- Router has seen its own Router ID in the Neighbor field of the neighbor’s Hello packets.

**Explicit Acknowledgment**

- A LSAck packet containing the LSA header is received.

**Implicit Acknowledgment**

- An Update packet that contains the same instance of the LSA.

**LSA Selection**

- Compare the seq, highest is more recent.
- The LSA with the highest unsigned checksum is the more recent.

If the ages of the LSAs differ by more than 15 minutes (MaxAgeDiff), the LSA with the lower age is more recent, but MaxAge (600 seconds) is more recent.

**Flooding**

- The router sends DD packets.

**DR/BDR Election**

- Highest priority wins (0-255); 0-do not participate, 1-default. Highest RID wins if Priority is the same.

**DR and BDR reach full state**, but DROther stops at 2Way with each other – no need to proceed to DD exchange as DR/BDR is elected.

- DR limits flooding and generates LSAs containing shared routes.

**Authentication**

- Authentication is checked when forming adjacency. All routers in area must be enabled for authentication (if per-area authentication is used), but not all links must have password set (only link which need to be protected).

- All routers within an area are not required to have authentication enabled if per-interface authentication is used.

**DR and BDR reach full state**, but DROther stops at 2Way with each other – no need to proceed to DD exchange as DR/BDR is elected.

- DR limits flooding and generates LSAs containing shared routes.

**Authentication**

- Authentication is checked when forming adjacency. All routers in area must be enabled for authentication (if per-area authentication is used), but not all links must have password set (only link which need to be protected).

- All routers within an area are not required to have authentication enabled if per-interface authentication is used.

**Authentication**

- Authentication is checked when forming adjacency. All routers in area must be enabled for authentication (if per-area authentication is used), but not all links must have password set (only link which need to be protected).

- All routers within an area are not required to have authentication enabled if per-interface authentication is used.

**Area <id> virtual-link <rid> authentication (null | authentication-key key-value | authentication-key key-value | authentication-message-digest key-value | authentication-message-digest key-value) null**

**Explicit Acknowledgment**

- A LSAck packet containing the LSA header is received.

**Implicit Acknowledgment**

- An Update packet that contains the same instance of the LSA.

**The LSA is retransmitted every 30 minutes until ACKed or adjacency is down.**

**LSUs containing retransmissions are always unicast, regardless of the network type.**

**Direct ACK**

- When duplicate LSA is received from a neighbor.

**When LSA’s age is MaxAge and receiving router does not have that LSA.**

**Loading**

- Router sends LSU packets.
**OSPF LSAs**

**LSA1: Router**
- Describes router interfaces in an area. Lists neighboring routers on each interface. LSID = RID
- Originated only by DR

**LSA2: Network**
- Describes transit networks for which DR has been elected
- LSID = DR's interface address
- Originated only by DR

**LSA3: Net summary**
- If an ABR knows multiple routes to destination within own area, it originates a single LSA3 into backbone with the lowest cost of the multiple routes.

**LSA4: ASBR Summary**
- ABR closest to ASBR creates LSA4 - cost to ASBR
- Not generated in NSSA, as FA is already set to ASBR
- ABR sends LSA3 with LSA1 and LSA2 subnets (simple vector - net, mask ABR's cost to reach that net)

**LSA5: AS External**
- If one network changes inside one area all routers in this area perform full SPF calculation, but outside that area, only cost is updated by ABR (partial SPF is run but other area routers)

**LSA6: Group membership**
- Created by ABR within NSSA area. LS4 is not generated by ABR for ASBR, as FA is used in place of LS4

**LSA7: Nssa External**
- For E2, simple LS10 is created and flooded into all areas
- For E1 routers in different areas perform 3-way calculation: Cost to ABR (LSA1) + Cost to ASBR (LSA4) + cost of E1 route

**LSA8: External Attributes LSA**
- Encapsulates ABR's external Type 1 (E1) metric calculation

**LSA9: Opaque LSA (link-local scope)**
- Carries FA pointing to external route source ASBR if external link is broadcast of non-broadcast. FA must be in routing table to be used by routers, so external link, usually pointing to NH (FA) must be enabled for OSPF network statement to be advertised

**LSA10: Opaque LSA (area-local scope)**
- Ignore lsa ospf!

**LSA11: Opaque LSA (AS scope)**
### Path attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flags</td>
<td>Code</td>
<td></td>
</tr>
<tr>
<td>0 – 15byte</td>
<td>1 – 2bytes (Attr Len Field)</td>
<td></td>
</tr>
<tr>
<td>0 – Non-transitive</td>
<td>1 – Transitive</td>
<td></td>
</tr>
<tr>
<td>0 – Well-known</td>
<td>1 – Optional</td>
<td></td>
</tr>
</tbody>
</table>

| 1 | Origin | WK M |
| 2 | AS_Path | WK M |
| 3 | Next_Hop | WK M |
| 4 | MED | O NT |
| 5 | Local_Pref | WK D |
| 6 | Atomic_Aggregate | WK D |
| 7 | Aggregator | O T |
| 8 | Community | O T |
| 9 | Originator_ID | O NT |
| 10 | Cluster | List |
| 11 | Advertiser | |
| 12 | RD/G_Paths/Cluster_Id | |
| 13 | MP-reachable NLRI | O NT |
| 14 | MP-unreachable NLRI | O NT |
| 15 | Extended Communities | |

### Decision Process

1. **Largest Weight** (locally originated paths: 32768, other 0)
2. **Largest Local-Preference** (Bgp default local preference) default 100
3. **Shortest AS_PATH** (in global, network, redistribute, aggregate)
4. **Lowest origin code** (bgp 0-IGP, bgp 1-EGP, bgp 2-Incomplete)
5. **Closest IGP neighbor** (best cost)
6. **Lowest Router-ID**
7. **Minimum Cluster-List length** (RR environment)
8. **Lowest Router-ID**
9. **Minimum Cluster-List length** (RR environment)
10. **If paths are external choose the oldest one** (flap prevention). Skipped if "bgp bestpath compare-routerid"

### RegExp

- Single character
- Zero or more
- One or more
- Zero or one
- Range
- Negate range
- Beginning of input
- End of input
- ( ) * . , { } ( ) ^ $, space
- Escape special character
- Repeat a match in ( )
- Logical OR

### BGP

- **bgp scan-time**<scanner-interval>
- **timers bgp**<keepalive>|<hold>|<min-hold>
- **neighbor ip**
- **enable**
- **keepalive**
- **holdtime**
- **TCP path MTU discovery** enabled by default for all BGP neighbor sessions
- **TCP path MTU discovery**
- **CFG IP MTU discovery**

### Features

- **IDLE**
- **OPEN-SENT**
- **OPEN-CONFIRM**
- **ESTABLISHED**

### Timers

- **TCPKeepAlive** every 60 sec (19 bytes header); Holdtime 180 sec
- **Keepalive**
- **TCPKeepAlive**

### Security

- **MD5 authentication**
- **Password**
- **TTL Security hops**
- **Packet TTL**
- **Packet TTL security hops**

### Paths

- **Origin**
- **WP**
- **AS Path**
- **WP**
- **Next Hop**
- **WP**
- **MED**
- **WP**
- **Local Pref**
- **WP**
- **Atomic Aggregate**
- **WP**
- **Aggregator**
- **WP**
- **Cluster List**
- **WP**

---

*By Krzysztof Zaleski, CCIE #24081. This Booklet is available for free and can be freely distributed in a form as is. Selling is prohibited.*
BGP Attributes

**AS_PATH**
- Stores information about the Autonomous System that passed the route from its origin to its current location.
- Enables route filtering and summarization.
- Used in configuring prefix lists.

**Community**
- Identifies routes from a specific AS.
- Used in route filtering and summarization.
- Allows for fine-grained control over route visibility.

**Cost**
- Represents the cost associated with a route.
- Used in route selection algorithms.
- Can be used to influence the selection of the best path.

**Local_Pref**
- Represents a local preference value.
- Used to influence the selection of the best path within the same AS.

**MED**
- Represents a Multi-Exit Discriminator.
- Used to influence the selection of the best path between ASes.

**Next_Hop**
- Represents the next hop IP address.
- Used to determine the path to the destination.

**ORIGIN**
- Represents the origin of a route.
- Used to determine if a route is directly connected, RIP, IGP, or EGP.

**Weight**
- Represents a weight assigned to a route.
- Used in route selection algorithms.

**Set Commands**
- Allows for route configuration.
- Can be used to add, remove, or set specific attributes.

**Route Tag**
- Stores information about the route tag.
- Used in route filtering and summarization.

**Set Local Preference**
- Sets the local preference value.
- Used to influence the selection of the best path within the same AS.

**Set MED**
- Sets the Multi-Exit Discriminator.
- Used to influence the selection of the best path between ASes.

**Set Next Hop**
- Sets the next hop IP address.
- Used to determine the path to the destination.

**Set Origin**
- Sets the origin of a route.
- Used to determine if a route is directly connected, RIP, IGP, or EGP.

**Set Weight**
- Sets the weight assigned to a route.
- Used in route selection algorithms.

**Route Map**
- Associates a route map with a route.
- Used to apply set commands to a route.

**Set Local Preference**
- Sets the local preference value.
- Used to influence the selection of the best path within the same AS.

**Set MED**
- Sets the Multi-Exit Discriminator.
- Used to influence the selection of the best path between ASes.

**Set Next Hop**
- Sets the next hop IP address.
- Used to determine the path to the destination.

**Set Origin**
- Sets the origin of a route.
- Used to determine if a route is directly connected, RIP, IGP, or EGP.

**Set Weight**
- Sets the weight assigned to a route.
- Used in route selection algorithms.
An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.

An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4.

Static

No sanity check like in IPv4, because neighbours use Link-Local IP addresses.
**IPv6 Tunneling**

IPv6 Tunneling can be used to connect IPv6 networks that are not directly connected. There are several types of IPv6 tunnels:

- **tunnel mode ipv6ip**
  - The same idea as GRE, but less overhead (IPv6 => IPv4).
  - Protocol number is 41.
  - Cisco recommends ISATAP instead of this. ::/96 used in a form of ::A.B.C.D where A.B.C.D is IPv4 address.
  - Destination automatically derived from tunnel interface address.
  - Trick to translate source IP from IPv4 to IPv6!!!

- **tunnel mode gre ipv6**
  - IPv6 over IPv6 GRE. Tunnel with source and destination IPv6 addresses. It’s not IPv6 over IPv4 tunneling.
  - In GRE IP, and IPv6IP Tunnel source and destination is IPv4 address. Both tunnels are p2p in nature (only). tunnel source is IPv4 address and tunnel destination is IPv6 address.
  - Plain GRE can be used, but much more overhead (ipv6 => GRE => IPv4). Protocol number is 47.
  - Also, GRE can send dynamic routing protocols which do not work on IPv4 directly. Protocols like ISIS require GRE tunneling.

- **IPv4-compatible**

- **Manual Automatic**
  - 6to4
  - Tunnel destination SHOULD NOT be configured. It is automatically determined per-each-packet.
  - Only one such tunnel allowed on device.
  - Trick to translate source IP from IPv4 to IPv6!!!
  - (G) ipv6 general-prefix <name> 6to4 loopback 0 show ipv6 general-prefix

- **ISATAP**
  - Host sends router discovery packet to a router to find an IPv6 prefix. It construts own address:
    - [64-bit link-local or globalunicast prefix]:0000:5efe:[IPv4 address of ISATAP link]
    - RA is disabled on tunnel interfaces, but it is required by ISATAP.
    - ISATAP uses IPv4 as a virtual NBMA data link layer, so it does not require IPv4 network infrastructure to support multicast. Supports point-to-multipoint communication.
    - ISATAP hosts must be configured with a potential routers list (PRL). PRL is typically built by consulting the DNS. Host configured for ISATAP asks DNS for remote router’s IP serving as IPv4/IPv6 endpoint.
  - Requires special address reserved for 6to4 (2002::/16) 2002:border-router-IPv4-address::/48.
  - Tunnel source loopback0 tunnel mode ipv6ip 6to4 ipv4 route 2002::/16 tunnel0 (required).
  - Trust to translate source IP from IPv4 to IPv6!!!
  - (G) ipv6 general-prefix <name> 6to4 loopback 0 show ipv6 general-prefix

- **NAT-PT**
  - In IPv4 NAT both source and destinations must always be translated. Cisco highly recommends NOT to use NAT-PT since it will be probably obsoleted.
  - NAT-PT uses IPv4 as a virtual NBMA data link layer, so it does not require IPv4 network infrastructure to support multicast. Supports point-to-multipoint communication.
  - Most sends router discovery packet to find an IPv6 prefix. It construts own address: [84-bit link-local or globalunicast prefix]:0000:Safe [IPv4 address of ISATAP link]
  - Requires special address reserved for 6to4 (2002::/16) 2002:border-router-IPv4-address::/48.
  - Tunnel source loopback0 tunnel mode ipv6ip 6to4 ipv4 route 2002::/16 tunnel0 (required).
  - RT A: (G) ipv6 route 2002::/16 tunnel0 (required)
224.0.0.0 – 239.255.255.255 (1110) 224.0.0.0/24 – Link local 224.0.1.0/24 – IANA assigned 224.0.1.0/24 => A4 | 8B | 0x0 =| 164 | 139 232.0.0.0/8 – SSM 233.0.0.0/8 – GLOP (public AS to Mcast) AS42123 => 42123 => 164 | 139 239.0.0.0/8 – Administrively scoped

**General rules**

- **Interface with lowest cost/metric to S or RP is choosen in calculating RPF. Highest intf IP wins if costs are the same.**
- **RPF interface is calculated for every Mcast entry every 5 sec.**
- **When a new neighbour is added to interface, the interface is reset to Forward/Dense state in all (*,G). New neighbor receives multicast instantly so it can create own (*,G) and (S,G) entries.**
- **Sparce or Dense mode specifies which groups can be send to the interface. The interface accepts ALL groups, regardless of mode.**

**PIM**

**Designated Router**

- Elected on every shared segment
- *ip pim dr-priority #* - Highest Priority (default 1) or IP. New router with higher priority/IP preempts existing DR.
- *ip pim dr-priority #* - No meaning for PIM-DM
- *ip pim register-state <name>* - Responsible for sending joins to S for receivers on the segment and Register messages to RP for active sources on the segment

**Neighbor**

- *ip pim dr-priority #* - Higest Priority (default 1) or IP. New router with higher priority/IP preempts existing DR.
- *ip pim dr-priority #* - No meaning for PIM-DM
- *ip pim register-state <name>* - Responsible for sending joins to S for receivers on the segment and Register messages to RP for active sources on the segment

**RPF check**

- May fail if Mcast stream is received on interface which is not enabled for Mcast.
- Interface with lowest cost/metric to S or RP is choosen in calculating RPF. Highest intf IP if costs are the same.

**Solution to RPF failure**

- May be a static minute (not really a route - it says that it is Ok. to receive Mcast from SRC from specified neighbour - overriding RPF)
- May result in multicast group(s) not accepted by interface. The interface receives ALL groups, regardless of mode.

**PIM Assert**

- If a router receives a PIM Assert message which is better, it removes (S/G) state from outgoing interface and stops flooding traffic.
- If a router receives a PIM Assert message which is worse, it initiates own PIM Assert message to inform the other router to stop flooding traffic.

**PIM Assert message**

- Originated by a router which is not a PIM Sparse Mode Designated Router (PIM-DM DR).
- Contains information about the senders and receivers on the shared segment.
**PIM-SM**

- **Register**
  - Unicast to RP with encapsulated Mcast packets
  - RP pims SPT if receivers are present
  - If no receivers are registered
  - Flags are set in (S,G)
  - RP sends a Register Stop
  - RP stops sending Unicast Registers
  - Source border router starts 1 min Register Suppression timer and then tries again 5 sec before expiration with Null-Register, if no register-stop is received full Register is sent

- **RP**
  - RP joins SPT if receivers are present

- **Ucast**
  - Sent by RP when starts receiving Mcast for (S,G) or automatically if no receivers are registered
  - Ucast to RP with encapsulated Mcast packets

- **NBMA**
  - (IF) ip pim nbma-mode
    - Separate peers next-hop is maintained in (*,G) and (S,G) OILs

- **Flexible Switching**
  - Every (S,G) J-flagged entry is calculated every 1 minute to see if traffic rate is below threshold, so it can switch back to RPT
  - ip pim stp-threshold immediate | <kb>
  - If kb is 0, then switchover is immediate (J-flag always present). Calculated every second

- **Source Registration**
  - Switchover takes place on last-hop router (closest to the receiver)
  - DR sends SPT-specific Join to S (derived from first Mcast packet), and sends RP-bit Prune to RP
  - Receivers connected to SPT on the way between RP and S join that tree immediately without going to RP
  - If rate is exceeded, J-flag is set in (*,G)
  - Every (S,G) J-flagged entry is calculated every 1 minute to see if traffic rate is below threshold, so it can switch back to RPT

- **PIM Sparse Mode**
  - OIL list of (*,G) reflects interfaces where (1) neighbors exist, (2) directly connected clients exist
  - Outgoing intf is not deleted upon receiving Prune. It is marked as Prune/Dense for 3 minutes. Then set back to Forward/Dense
  - Prune-override — router waits 3 sec for Join from another router on shared LAN

- **Rendezvous Point**
  - Auto-RIP (PIMv1, Cisco proprietary)
  - Bootstrap (PIMv2, standardized)

**PIM-Dense Mode**

- **Pruning**
  - (*,G) Prune is sent to upstream router, which in turn removes interface from OIL. Process is repeated toward RP. Prunes are sent immediately, but entries with P-flag set are deleted after 3-min timeout
  - ip pim dm-fallback
  - Any group for which RP does not exist automatically switches by default back to DM

- **Rules**
  - Allows Auto-RP dense-mode groups 224.0.1.39 and 224.0.1.40 to be distributed while using sparse-mode groups
  - ip pim state-refresh disable
  - State-refresh is enabled by default. Unsolicited Prune sent every 60 sec. toward upstream router, so no periodic flood is required. Keeps pruned state on branches with no receivers

**PIM-Dense Mode Operations**

- **Register**
  - RP pims SPT if receivers are present

- **Register Stop**
  - RP stops sending Unicast Registers

- **Source border router**
  - Starts 1 min Register Suppression timer and then tries again 5 sec before expiration with Null-Register, if no register-stop is received full Register is sent

**PIM Sparse Mode Operations**

- **Register**
  - RP pims SPT if receivers are present

- **Register Stop**
  - RP stops sending Unicast Registers

- **Source border router**
  - Starts 1 min Register Suppression timer and then tries again 5 sec before expiration with Null-Register, if no register-stop is received full Register is sent

- **Filtering**
  - Prevent unwanted RPs or mcast groups to become active in SM domain. Must be configured on every router.
  - ip pim accept-register (list | catch | route-map)
  - Extended ACLs used for multicast filtering (any) is used as follows:
    - access-list 100 permit (source ip | wildcard | group address | wildcard)

- **Rendezvous Point**
  - Auto-RIP (PIMv1, Cisco proprietary)
  - Bootstrap (PIMv2, standardized)

- **Pruning**
  - (*,G) Prune is sent to upstream router, which in turn removes interface from OIL. Process is repeated toward RP. Prunes are sent immediately, but entries with P-flag set are deleted after 3-min timeout

- **Rules**
  - Allows Auto-RP dense-mode groups 224.0.1.39 and 224.0.1.40 to be distributed while using sparse-mode groups
  - ip pim state-refresh disable
  - State-refresh is enabled by default. Unsolicited Prune sent every 60 sec. toward upstream router, so no periodic flood is required. Keeps pruned state on branches with no receivers

**PIM-Dense Mode Operations**

- **Register**
  - RP pims SPT if receivers are present

- **Register Stop**
  - RP stops sending Unicast Registers

- **Source border router**
  - Starts 1 min Register Suppression timer and then tries again 5 sec before expiration with Null-Register, if no register-stop is received full Register is sent

**PIM Sparse Mode Operations**

- **Register**
  - RP pims SPT if receivers are present

- **Register Stop**
  - RP stops sending Unicast Registers

- **Source border router**
  - Starts 1 min Register Suppression timer and then tries again 5 sec before expiration with Null-Register, if no register-stop is received full Register is sent
Auto-RP

(Cisco)

Mapping Agent

Candidate RP

Features

Cisco-RP Announce sent to 224.0.1.39 UDP/496
Every 60 sec with holdtime 180 sec

If ACL is not defined whole Most Range is included. Do not use deny
statement in C-RP ACLs. Only contiguous masks are allowed in group ACL.

Multiple C-RPs may exist for G. Highest IP is selected by Mapping agent.

All routers join Cisco-RP Discovery 224.0.1.40 to learn mappings from MA

Cisco-RP Announce sent to 224.0.1.39 UDP/496

Messages sent to UDP/496 every 60 sec with holdtime 180 sec.

There can be many MAs (independent) for different groups, but for the same
group, the one with highest IP wins, and the rest cease their announcements.

Every 60 sec with holdtime 180 sec.

ip pim send-rp-announce <if> scope <ttl> [group-list <acl>] [interval <sec>]
If ACL is not defined whole Mcast range is included. Do not use deny
statement in C-RP ACLs. Only contiguous masks are allowed in group ACL.

ip pim send-rp-discovery scope <ttl> [interval <sec>]

Router joins 224.0.1.39 (becomes G member), and sends mappings to 224.0.1.40

There can be many MAs (independent) for different groups, but for the same

C-RP with highest IP is announced for the same range. If one range
is a subset of another, but RPs are different, both are announced.

Group ACL is used to override default. 

ip pim announce-filter <if-list> [group-list <acl2>]

Avoid spoofing (Allowed RPs in ACL1 for groups in ACL2) - ONLY on mapping agent.

Features

ip pim sparse-dense-mode

is required

ip pim rp-set RP range to RP-set mapping

OSPF is used to announce RP to RPF neighbors.

Hashing

ip pim rp-candidate <if> [group-list <acl>] interval <sec> group-list <acl> priority <#>

Highest hash for a group range wins. If it's the same then highest IP wins.

All routers perform the same hashing to select RP for specific G.

Mask defines how many consecutive Gs will be hashed to one RP

Hash is calculated from C-RP, G, and mask

IP pim bar-candidate loopback <31>

If there are two RPs, the load will be evenly distributed among them.
Switch detects router port by listening for IGMP Query. OSPF, HSRP, PIMv2, DVMRP:
- Only IGMP messages are processed by switch CPU.
- Only switch is Switch involved.
- 2. CPU floods to all ports.
- 3. No suppression. CPU intercepts all Reports.
- 4. IGMP report creates CAM entry with ports Host + Router + CPU.
- 5. Leave is intercepted by CPU.

No suppression. CPU intercepts all Reports:
- Only IGMP messages are processed by switch CPU.
- Only switch is Switch involved.
- 2. CPU sends General Query on host's port to see if there are other hosts
- 3. If no more hosts port is removed from CAM
- 4. CPU sends Leave to router if no CAM entries
- 5. One Report sent to router by CPU.

Snooping:
- CPU sends Leave to router if no CAM entries
- 5. One Report sent to router by CPU.

Join:
- 1. Host sends IGMP Join to R
- 2. R calculates Mcast MAC (GDA) from IP Mcast sent by host
- 3. R sends IGMP Join to CGMP MAC
- 4. Switch creates Mcast CAM with R port
- 5. Switch gets host's USA MAC and adds port to Mcast CAM

CGMP:
- Only router sends CGMP, and switch only intercepts
- Router reports itself to switch every 60 sec (GDA = 0.0.0.0 USA = router MAC)
- If source-only is detected R sends CGMP Join with own USA, so CAM is created for G (no flooding).

CGMP Query:
- 1. router's Query is intercepted by CPU
- 2. CPU sends General Query on host's port to see if there are other hosts
- 3. If no more hosts port is removed from CAM
- 4. CPU sends Leave to router if no CAM entries
- 5. One Leave sent to router by CPU

Querier – Router with lowest IP (for IGMP v.2) and v.3. for IGMP v.1 DR is elected using PIM on multiaccess network, responsible for sending membership queries to the LAN, and building shared trees

Querier Present Interval = 256 (2x General Q Int + ½ of Q Response int)

Backup querier becomes an active one if does not hear queries from the other router within this amount of time

Join:
- 1. Host sends IGMP Join to R
- 2. R calculates Mcast MAC (GDA) from IP Mcast sent by host
- 3. R sends IGMP Join to CGMP MAC
- 4. Switch creates Mcast CAM with R port
- 5. Switch gets host's USA MAC and adds port to Mcast CAM

Join Group:
- (IF) ip igmp join-group <group> [source <src IP>]
- (IF) ip igmp static-group
- (IF) ip igmp filter <id>
- (G) ip igmp limit <#>
- (IF) ip igmp access-group <name>
- (IF) ip igmp profile <id>
- (IF) ip igmp max-gr <#>
- (G) ip igmp limit <#> [except <acl>]
- (G) ip igmp access-group <name>
- (G) ip igmp profile <id>
Multicast VLAN registration intercepts IGMP Joins

Based only on shared tree, no STP (many to many, receivers are also senders)

Source sends traffic unconditionally to RP at any time (no registration process like in SM)

Designated forwarder (PIM assert) is used on each link for loop prevention

No (S,G) entries, only (*,G) mroute states are active

Traffic may flow up and down the tree

ip pim bidir-enable

RP can be set manually, with BSR or Auto-RP. For the the automatic methods, a BSR keyword is required at the end (send-bsr-announce and rp-candidate)

Does not require RP (no shared trees). Only Source trees are built

Only edge routers must support SSM, other routers only require PIM-SM

ip igmp version 3

Requires IGMPV3 (INCLUDE/EXCLUDE messages). Hosts can decide which sources they want to join explicitly. The (*,G) joins are dropped.

ip pim ssm (default) | range <acl>

Enable SSM for either default SSM range (232.0.0.0/8), or only for ranges defined in ACL

Source discovery is not a part of SSM. Other means must be implemented to support source discovery

Filtering

PIM Register messages cannot be filtered with this feature

If filter-autorp option is used, then all groups from Auto-RP announcements and discoveries are removed, if they do not match the ACL. If any part of the group is denied, then whole announced range is denied.

Multicast helper for broadcast traffic

Forward broadcast sent to UDP/5555 from one LAN segment to another using Mcast

Not all UDP broadcast can be automatically forwarded. To enable additional UDP port ip forward protocol <port number> must be added on all edge routers.

No PIM adjacency

PIM adjacency

Multicast helper for broadcast traffic

Broadcast Sender

Change broadcast to multicast

interface serial 0/0

ip pim sparse-dense-mode

ip pim neighbor-filter 1

access-list 1 deny 10.0.0.2

interface fastethernet 0/0

ip pim sparse-dense-mode

ip igmp helper-address 10.0.0.1

IGMP Join

Multicast VLAN registration intercepts IGMP Joins

Always subscribe on a port to subscribe to a multicast stream on the network-wide multicast VLAN.

Single multicast VLAN can be shared on the network while subscribers remain in separate VLANs

Multicast routing and MVR cannot coexist on a switch

 enrge MVR for a group and # of consecutive groups (max 256). Groups should not be aliasing (1:32 ratio)

ip mvr <ip> [<count>]

Define which VLAN carries actual multicast traffic

ip multicast rate-limit {in | out} [group-list <acl>] [source-list <acl>] [<kbps>]

Only edge routers must support SSM, other routers only require PIM-SM

ip igmp helper-address <hub’s WAN IP>

Multicast must be enabled on each interface, as mcast traffic can be flooded, but filtering must be used, so hub does not form PIM adjacency to spoke, so no automatic flooding is performed (in dense-mode)

ip pim neighbor-filter <acl>

Configured on hub’s WAN interface. ACL must have only deny statement for spoke’s WAN IP. Hub router drops Hellos from spoke, but spoke accepts hellos and sees the hub neighbor.

ip mvr vlan <id>

Desigranted forwarder (PIM assert) is used on each link for loop prevention

Bidir PIM

Multicast routing and MVR cannot coexist on a switch

Traffic may flow up and down the tree

ip pim assert-enable

Desigranted forwarder (PIM assert) is used on each link for loop prevention

PIM Assert

By default all mcast enabled interfaces have TTL 0 – TTL in mcast packet must be higher than configured on interface

ip multicast ttl-threshold <#>

Uses IGMP v3 messages to carry routing information. Metric is a hop-count like in RIP

Router sends periodic reports with a list of directly connected subnets

Routes received via DVMRP are only used for RPF, not for directing traffic toward destination

ip dvmrp unicast-routing

Not fully implemented on IOS. Can be enabled only on edge routers and interfaces peer with DVMRP-capable legacy devices

ip dvmrp unicast-routing

Enables DVMRP routes to take precedence over unicast routes for checking RPF

ip dvmrp metric <#> [list <acl>] [protocol <process id>]

By default router will advertise only connected subnets. Other subnets can also be advertised, with assigned metric (0 means do not advertise). If protocol is not defined metric is set only for connected subnets.

ip pim sparse-dense-mode

Like in RIP, routes are automatically summarized

stub router

By default route is advertised to router on connected subnets

ip pim sparse-dense-mode

Forward broadcast sent to UDP/5555 from one LAN segment to another using Mcast

Not all UDP broadcast can be automatically forwarded. To enable additional UDP port ip forward protocol <port number> must be added on all edge routers.

ip igmp helper-address <hub’s WAN IP>

Multicast must be enabled on each interface, so mcast traffic can be flooded, but filtering must be used, so hub does not form PIM adjacency to spoke, so no automatic flooding is performed (in dense-mode)

ip pim neighbor-filter <acl>

Configured on hub’s WAN interface. ACL must have only deny statement for spoke’s WAN IP. Hub router drops Hellos from spoke, but spoke accepts hellos and sees the hub neighbor.

ip mvr vlan <id>

Desigranted forwarder (PIM assert) is used on each link for loop prevention

Bidir PIM

Multicast routing and MVR cannot coexist on a switch

Traffic may flow up and down the tree

ip pim assert-enable

Desigranted forwarder (PIM assert) is used on each link for loop prevention

PIM Assert

By default all mcast enabled interfaces have TTL 0 – TTL in mcast packet must be higher than configured on interface

ip multicast ttl-threshold <#>

Uses IGMP v3 messages to carry routing information. Metric is a hop-count like in RIP

Router sends periodic reports with a list of directly connected subnets

Routes received via DVMRP are only used for RPF, not for directing traffic toward destination

ip dvmrp unicast-routing

Not fully implemented on IOS. Can be enabled only on edge routers and interfaces peer with DVMRP-capable legacy devices

ip dvmrp unicast-routing

Enables DVMRP routes to take precedence over unicast routes for checking RPF

ip dvmrp metric <#> [list <acl>] [protocol <process id>]

By default router will advertise only connected subnets. Other subnets can also be advertised, with assigned metric (0 means do not advertise). If protocol is not defined metric is set only for connected subnets.

ip pim sparse-dense-mode

Like in RIP, routes are automatically summarized

stub router

By default route is advertised to router on connected subnets

ip pim sparse-dense-mode

Forward broadcast sent to UDP/5555 from one LAN segment to another using Mcast

Not all UDP broadcast can be automatically forwarded. To enable additional UDP port ip forward protocol <port number> must be added on all edge routers.

ip igmp helper-address <hub’s WAN IP>

Multicast must be enabled on each interface, so mcast traffic can be flooded, but filtering must be used, so hub does not form PIM adjacency to spoke, so no automatic flooding is performed (in dense-mode)

ip pim neighbor-filter <acl>

Configured on hub’s WAN interface. ACL must have only deny statement for spoke’s WAN IP. Hub router drops Hellos from spoke, but spoke accepts hellos and sees the hub neighbor.

ip mvr vlan <id>

Desigranted forwarder (PIM assert) is used on each link for loop prevention

Bidir PIM

Multicast routing and MVR cannot coexist on a switch

Traffic may flow up and down the tree

ip pim assert-enable

Desigranted forwarder (PIM assert) is used on each link for loop prevention

PIM Assert

By default all mcast enabled interfaces have TTL 0 – TTL in mcast packet must be higher than configured on interface

ip multicast ttl-threshold <#>

Uses IGMP v3 messages to carry routing information. Metric is a hop-count like in RIP

Router sends periodic reports with a list of directly connected subnets

Routes received via DVMRP are only used for RPF, not for directing traffic toward destination

ip dvmrp unicast-routing

Not fully implemented on IOS. Can be enabled only on edge routers and interfaces peer with DVMRP-capable legacy devices

ip dvmrp unicast-routing

Enables DVMRP routes to take precedence over unicast routes for checking RPF

ip dvmrp metric <#> [list <acl>] [protocol <process id>]

By default router will advertise only connected subnets. Other subnets can also be advertised, with assigned metric (0 means do not advertise). If protocol is not defined metric is set only for connected subnets.

ip pim sparse-dense-mode

Like in RIP, routes are automatically summarized

stub router

By default route is advertised to router on connected subnets

ip pim sparse-dense-mode

Forward broadcast sent to UDP/5555 from one LAN segment to another using Mcast

Not all UDP broadcast can be automatically forwarded. To enable additional UDP port ip forward protocol <port number> must be added on all edge routers.

ip igmp helper-address <hub’s WAN IP>

Multicast must be enabled on each interface, so mcast traffic can be flooded, but filtering must be used, so hub does not form PIM adjacency to spoke, so no automatic flooding is performed (in dense-mode)

ip pim neighbor-filter <acl>

Configured on hub’s WAN interface. ACL must have only deny statement for spoke’s WAN IP. Hub router drops Hellos from spoke, but spoke accepts hellos and sees the hub neighbor.

ip mvr vlan <id>

Desigranted forwarder (PIM assert) is used on each link for loop prevention

Bidir PIM

Multicast routing and MVR cannot coexist on a switch

Traffic may flow up and down the tree

ip pim assert-enable

Desigranted forwarder (PIM assert) is used on each link for loop prevention

PIM Assert

By default all mcast enabled interfaces have TTL 0 – TTL in mcast packet must be higher than configured on interface

ip multicast ttl-threshold <#>

Uses IGMP v3 messages to carry routing information. Metric is a hop-count like in RIP

Router sends periodic reports with a list of directly connected subnets

Routes received via DVMRP are only used for RPF, not for directing traffic toward destination

ip dvmrp unicast-routing

Not fully implemented on IOS. Can be enabled only on edge routers and interfaces peer with DVMRP-capable legacy devices

ip dvmrp unicast-routing

Enables DVMRP routes to take precedence over unicast routes for checking RPF

ip dvmrp metric <#> [list <acl>] [protocol <process id>]

By default router will advertise only connected subnets. Other subnets can also be advertised, with assigned metric (0 means do not advertise). If protocol is not defined metric is set only for connected subnets.

ip pim sparse-dense-mode

Like in RIP, routes are automatically summarized

stub router

By default route is advertised to router on connected subnets

ip pim sparse-dense-mode

Forward broadcast sent to UDP/5555 from one LAN segment to another using Mcast

Not all UDP broadcast can be automatically forwarded. To enable additional UDP port ip forward protocol <port number> must be added on all edge routers.

ip igmp helper-address <hub’s WAN IP>

Multicast must be enabled on each interface, so mcast traffic can be flooded, but filtering must be used, so hub does not form PIM adjacency to spoke, so no automatic flooding is performed (in dense-mode)

ip pim neighbor-filter <acl>

Configured on hub’s WAN interface. ACL must have only deny statement for spoke’s WAN IP. Hub router drops Hellos from spoke, but spoke accepts hellos and sees the hub neighbor.

ip mvr vlan <id>
IPv6 Multicast

To enable IPv6 multicast routing on a router, you must first enable IPv6 unicast routing. IPv6 supports MLS, PIM-SM, and PIM-SSM. It does NOT support POM-DM.

Main concepts are exactly the same as for IPv4 (DR, BSR, RP, RPF).

BSR

- ipv6 pim bsr candidate bsr <ipv6-addr> [<hash>] [priority <val>]
  - Configures a router to be a candidate BSR. It will participate in BSR election.

- ipv6 pim bsr announced rp <ipv6-addr> [group-list <acl-name>] [priority <val>] [bidir] [scope <val>]
  - Announces scope-to-RP mappings directly from the BSR for the specified candidate RP of RP does not support BSR or is located outside company’s network. Normal RP announces mappings. Default priority is 192. The announced BSR mappings are announced only by the currently elected BSR.

- ipv6 pim bsr border
  - Configures a border for all BSMs of any scope.

- ipv6 pim spt-threshold infinity [group-list <acl-name>]
  - Configures when a PM leaf router uses the SPT for the specified groups (all groups if ACL=0).

- ipv6 pim hello-interval <sec>
  - Configures the frequency (30 sec default + small jitter) of PIM hello messages.

- ipv6 pim prune-interval <sec>
  - Configures periodic (90 sec default) PM prune announcements intervals.

Features

- ipv6 pim dr-priority <val>
  - Highest priority (default is 1) or highest IPv6 address becomes the DR for the LAN.

- (G) no ipv6 pim rp embedded
  - Embedded RP support allows the router to learn RP information using the multicast group destination address instead of the statically configured RP. Applies only to the embedded RP group ranges ffX::/16 and ffX::/16. Ex: FF7E:0140:2001:0DB8:C003:111D:0000:1112 => RP: 2001:0DB8:C003:111D::1

- (G) ipv6 pim rp-address <ipv6-address> [group-list <acl-name>]
  - Configures valid RP address for a particular group range.

- (G) ipv6 pim accept-register (list <acl> | route-map <name>)
  - Accepts or rejects registers at the RP. RP can be used to check BGP prefixes.

- (G) ipv6 pim rp-address <ipv6-address> [group-list <acl-name>]
  - Configures valid RP address for a particular group range.

- (IF) ipv6 pim hello-interval <sec>
  - Configures the frequency (30 sec default + small jitter) of PIM hello messages.

- (IF) ipv6 pim prune-interval <sec>
  - Configures periodic (90 sec default) PM prune announcements intervals.

Zones

- Each link, and the interfaces attached to that link, comprises a single zone of link-local scope.

- There is a single zone of global scope comprising all the links and interfaces in the Internet.

- The boundaries of zones of scope other than interface-local, link-local, and global must be defined and configured by network administrators.

- A zone is a particular instance of a topological region.

- A scope is the size of a topological region.

- Each link, and the interfaces attached to that link, comprises a single zone of link-local scope.

- The boundaries of zones of scope other than interface-local, link-local, and global must be defined and configured by network administrators.

- A zone of a given scope (less than global) falls completely within zones of larger scope; that is, a smaller scope zone cannot include more topology than any larger scope zone with which it shares any links or interfaces.

- A zone of a given scope (less than global) falls completely within zones of larger scope; that is, a smaller scope zone cannot include more topology than any larger scope zone with which it shares any links or interfaces.

- Each interface belongs to exactly one zone of each possible scope.

- (IF) ipv6 multicast boundary scope <value>
  - Configures a multicast boundary on the interface for a specified scope.

- show ipv6 pim interface (state-on) (state-off)
- show ipv6 pim (neighbor) [group-list]
- show ipv6 pim pruned interface [summary] [detail]
- show ipv6 pim range-list
- clear ipv6 pim (counter) [topology] [if]
- show ipv6 pim bsr [multicast] [rp-cache] [candidate-rp]
- show ipv6 pim multicast-routing
- show ipv6 pim dr-priority

Timers

- (G) ipv6 pim spt-threshold infinity [group-list <acl-name>]
  - Configures when a PM leaf router uses the SPT for the specified groups (all groups if ACL=0).

- (IF) ipv6 pim hello-interval <sec>
  - Configures the frequency (30 sec default + small jitter) of PIM hello messages.

- (IF) ipv6 pim prune-interval <sec>
  - Configures periodic (90 sec default) PM prune announcements intervals.

- (G) ipv6 pim rp-address <ipv6-address> [group-list <acl-name>]
  - Configures valid RP address for a particular group range.

- (G) ipv6 pim accept-register (list <acl> | route-map <name>)
  - Accepts or rejects registers at the RP. RP can be used to check BGP prefixes.

- (IF) ipv6 pimRP-border
  - Configures a border for all BSMs of any scope.

- show ipv6 pim interface (state-on) (state-off)
- show ipv6 pim (neighbor) [group-list]
- show ipv6 pim pruned interface [summary] [detail]
- show ipv6 pim range-list
- clear ipv6 pim (counter) [topology] [if]
- show ipv6 pim bsr [multicast] [rp-cache] [candidate-rp]
- show ipv6 pim multicast-routing
- show ipv6 pim dr-priority
MLD uses ICMPv6 to carry its messages. All MLD messages are link-local with a TTL=1. Router alert option is set.

**Features**

- **Query**: General - multicast address field is set to 0.
- **Group-specific and multicast-address-specific**: Multicast address is set to group address.
- **Report**: Multicast address field is set to specific IPv6 multicast address to which the host is listening.
- **Done**: Multicast address field is set to specific IPv6 multicast address to which the host was listening.

**Config**

- **ipv6 mld join-group [<group>] [include | exclude] {<source-ip> | source-list [<acl>]}
- **ipv6 mld access-group <ACL-name>
- **ipv6 mld static-group [<group>] [include | exclude] {<source-ip> | source-list [<acl>]}
- **ipv6 mld query-max-response-time <sec>
- **ipv6 mld query-timeout <sec>
- **ipv6 mld query-interval <sec>
- **ipv6 mld state-limit <#>
- **ipv6 mld limit <#> [except <acl>]
- **ipv6 mld explicit-tracking <ACL-name>
- **show ipv6 mld groups summary
- **show ipv6 mld interface [if]
- **show ipv6 mld traffic
- **clear ipv6 mld counters [if]"
Assured Forwarding AFxy => DSCP = 8*x + 2*y

Class-map match-any <name>
If ANY match statement within a class is matched, the class is executed

match ip prec 1 2 3
Any of specified IP Precedences needs to be matched (logical OR)

match protocol http mime image* - match all images
match protocol http mime image/jpeg
This would match jpeg,jpg,jpe,jfif,pjpeg, and pjp types

MPLS
3 bits MPLS Experimental field

Class Selector/IP is copied to Exp field in MPLS label

CEF required. Deep Packet Inspection – match difficult-to-match packets
match protocol http url “*important*”
match protocol http mime image* - match all images
match protocol fasttrack file-transfer * - match all P2P applications

Match CEF IP Precedence
3 bits IP Precedence (class selector) in TOS byte of IP header

Pre-classification
TOS field is copied from original field

NBAR
8 bits TOS byte in IP header
6 bits DSCP in TOS byte of IP header
TOS field is copied from original field

MQC
Up to 4 COS or IPP values can be set in one match cos/precedence statement
Up to 8 DSCP values can be set in one match dscp statement

Class-map match-all <name>
The class is executed only if ALL match statements within a class are matched

class-map match-traffic <name>
IP/Port match statement within a class is matched, the class is executed

match ip prec 1 2 3
Any of specified IP Precedences needs to be matched (logical OR)

FR DE
Policy-map class "<name>"

Legacy
Frame-relay de-list <#> protocol ip ...
Frame-relay de-group <#> <dlci>

L2
3 bits COS in 802 1Q VLAN frame. Possible only on trunk links, where 802 1Q tag and ISL encapsulation exist

3560
Treats IPv4 as non-IP traffic
Class-default is unpredictable, so additional class should be created to catch all IP and non-IP traffic (MAC ACL)
Max 64 queues/classes (63 + class-default)

- Only one variation of BW can be used (static or percentage)
- Static bandwidth configuration with BW assigned to class-default and not
- Bandwidth percent <%>
  - Always % of literal interface BW

- Interface bandwidth 100%
- Bandwidth remaining-percent %
  - % of reservable BW (% BW of total BW minus already reserved BW)

- Max reservable BW for non-class-default queues – 75%
  - If class-default has bandwidth defined it is also calculated as reservable

- Unallocated
- Weight = [32,384 / (IP Precedence + 1)]
- CDT (1-4096) has been reached, the packet may be thrown away
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
- CDT (1-4096) has been reached, the packet may be thrown away
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth

- There are two output queues. Software queue (FIFO, WFQ, CBWFQ), and hardware queue TX-ring. Software queue is filled only if hardware queue is full. Software queue does NOT kick in if there is no congestion

<table>
<thead>
<tr>
<th>Queue</th>
<th>BW</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>class A</td>
<td>20%</td>
<td>20</td>
</tr>
<tr>
<td>class B</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>class-default</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>class B</td>
<td>20%</td>
<td>20</td>
</tr>
<tr>
<td>class A</td>
<td>25%</td>
<td>25</td>
</tr>
<tr>
<td>class-default</td>
<td>25%</td>
<td>25</td>
</tr>
</tbody>
</table>

- FIFO within each queue except class-default (FIFO or WFQ)
- If CDT packets are already in the queue into which a packet should be placed, WFQ considers discarding the new packet, but if a packet with a larger SN has already been unqueued in a different queue, however, WFQ instead discards the packet with the larger SN

- Bandwidth limit <IP>
- Max packets per class (threshold for tail-drop). Default is 256.
- Power of 2 is accepted.
- WRED can be enabled on all queues (but not LLQ)
- Total 64 queues/classes (63 + class-default)
- Min packets per class (threshold for tail-drop). Default is 1

- CI’s output queue is always FIFO (default 75 packets)
- INTERFACE LIMITATION <BDT> (1-4096)

<table>
<thead>
<tr>
<th>Queue</th>
<th>BW</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>class A</td>
<td>20%</td>
<td>20</td>
</tr>
<tr>
<td>class B</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>class-default</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>class B</td>
<td>20%</td>
<td>20</td>
</tr>
<tr>
<td>class A</td>
<td>25%</td>
<td>25</td>
</tr>
<tr>
<td>class-default</td>
<td>25%</td>
<td>25</td>
</tr>
</tbody>
</table>

- Interface bandwidth 100%
- Bandwidth remaining-percent %
  - % of reservable BW (% BW of total BW minus already reserved BW)

- Unallocated
- Weight = [32,384 / (IP Precedence + 1)]
- CDT (1-4096) has been reached, the packet may be thrown away
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
- CDT (1-4096) has been reached, the packet may be thrown away
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth

- There are two output queues. Software queue (FIFO, WFQ, CBWFQ), and hardware queue TX-ring. Software queue is filled only if hardware queue is full. Software queue does NOT kick in if there is no congestion

- FIFO within each queue except class-default (FIFO or WFQ)
- If CDT packets are already in the queue into which a packet should be placed, WFQ considers discarding the new packet, but if a packet with a larger SN has already been unqueued in a different queue, however, WFQ instead discards the packet with the larger SN

- Bandwidth limit <IP>
- Max packets per class (threshold for tail-drop). Default is 256.
- Power of 2 is accepted.
- WRED can be enabled on all queues (but not LLQ)
- Total 64 queues/classes (63 + class-default)
- Min packets per class (threshold for tail-drop). Default is 1

- CI’s output queue is always FIFO (default 75 packets)
- INTERFACE LIMITATION <BDT> (1-4096)

<table>
<thead>
<tr>
<th>Queue</th>
<th>BW</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>class A</td>
<td>20%</td>
<td>20</td>
</tr>
<tr>
<td>class B</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>class-default</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>class B</td>
<td>20%</td>
<td>20</td>
</tr>
<tr>
<td>class A</td>
<td>25%</td>
<td>25</td>
</tr>
<tr>
<td>class-default</td>
<td>25%</td>
<td>25</td>
</tr>
</tbody>
</table>

- Interface bandwidth 100%
- Bandwidth remaining-percent %
  - % of reservable BW (% BW of total BW minus already reserved BW)

- Unallocated
- Weight = [32,384 / (IP Precedence + 1)]
- CDT (1-4096) has been reached, the packet may be thrown away
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
- CDT (1-4096) has been reached, the packet may be thrown away
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth

- There are two output queues. Software queue (FIFO, WFQ, CBWFQ), and hardware queue TX-ring. Software queue is filled only if hardware queue is full. Software queue does NOT kick in if there is no congestion

- FIFO within each queue except class-default (FIFO or WFQ)
- If CDT packets are already in the queue into which a packet should be placed, WFQ considers discarding the new packet, but if a packet with a larger SN has already been unqueued in a different queue, however, WFQ instead discards the packet with the larger SN

- Bandwidth limit <IP>
- Max packets per class (threshold for tail-drop). Default is 256.
- Power of 2 is accepted.
- WRED can be enabled on all queues (but not LLQ)
- Total 64 queues/classes (63 + class-default)
- Min packets per class (threshold for tail-drop). Default is 1

- CI’s output queue is always FIFO (default 75 packets)
- INTERFACE LIMITATION <BDT> (1-4096)

<table>
<thead>
<tr>
<th>Queue</th>
<th>BW</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>class A</td>
<td>20%</td>
<td>20</td>
</tr>
<tr>
<td>class B</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>class-default</td>
<td>15%</td>
<td>15</td>
</tr>
<tr>
<td>class B</td>
<td>20%</td>
<td>20</td>
</tr>
<tr>
<td>class A</td>
<td>25%</td>
<td>25</td>
</tr>
<tr>
<td>class-default</td>
<td>25%</td>
<td>25</td>
</tr>
</tbody>
</table>

- Interface bandwidth 100%
- Bandwidth remaining-percent %
  - % of reservable BW (% BW of total BW minus already reserved BW)

- Unallocated
- Weight = [32,384 / (IP Precedence + 1)]
- CDT (1-4096) has been reached, the packet may be thrown away
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
- CDT (1-4096) has been reached, the packet may be thrown away
- Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth

- There are two output queues. Software queue (FIFO, WFQ, CBWFQ), and hardware queue TX-ring. Software queue is filled only if hardware queue is full. Software queue does NOT kick in if there is no congestion

- FIFO within each queue except class-default (FIFO or WFQ)
- If CDT packets are already in the queue into which a packet should be placed, WFQ considers discarding the new packet, but if a packet with a larger SN has already been unqueued in a different queue, however, WFQ instead discards the packet with the larger SN

- Bandwidth limit <IP>
- Max packets per class (threshold for tail-drop). Default is 256.
- Power of 2 is accepted.
- WRED can be enabled on all queues (but not LLQ)
- Total 64 queues/classes (63 + class-default)
- Min packets per class (threshold for tail-drop). Default is 1

- CI’s output queue is always FIFO (default 75 packets)
- INTERFACE LIMITATION <BDT> (1-4096)
**TCP behaviour**

WRED still randomly picks the packet, but instead of discarding it, it marks a couple of bits in the packet header, and forwards the packet. Marking these bits begins a process which causes the sender to reduce CWND by 50%.

1) Both TCP endpoints agree that they can support ECN by setting ECN bits to either 01 or 10. If TCP sender does not support ECN, the bits should be set to 00. If ECN = 00 packet is discarded
2) Router checks the packet’s ECN bits, and sets the bits to 11 and forwards packet instead of discarding it.
3) TCP receiver notices ECN = 11 and sets Explicit Congestion Experienced (ECE) flag in the next TCP segment it sends back to the TCP sender.
4) TCP sender receives segment with ECE flag set and reduces CWND by half. TCP sender reduces CWND by half.
5) TCP sender sends Congestion Window Reduced (CWR) flag in next segment to inform receiver it slowed down.

**Configuration**

Can be configured only on main interfaces. Sets FIFO on interface

- random-detect - enable RED
- random-detect (dscp-based | prec-based)
- random-detect (dscp <dscp> | precedence <prec>) <min> <max> <mpd>
- random-detect exponential-weighting-constant <val>
- random-detect exponential-weighting-constant <val>
- random-detect flow random-detect flow count <flows>
- random-detect flow random-detect flow average-depth-factor <#>
- random-detect flow random-detect flow average-depth-factor <#>

**MPD**

Defines max discard percentage

MPD=5 => (1/MPD) * 100% => 1/5 * 100% = 20%

One out of 5 packets is dropped during congestion

**Average Queue Depth**

RED uses the average depth, and not the actual queue depth, because the actual queue depth will most likely change much more quickly than the average depth.

New average = (Old_average * (1 – 2^-n)) + (Current_Q_depth * 2^-n)

For default n=9 (EWC): New average = (Old_average * .998) + (Current_Q_depth * .002)

The average changes slowly, which helps RED prevent overreaction to changes in the queue depth. The higher the average the more steady WRED. Lower value reacts more quickly to avg depth changes.

**WRED**

The receiver grants the sender the right to send x bytes of data before requiring an acknowledgment, by setting the value x into the Window field of the TCP header.

The second window used by TCP is called the congestion window (CWND). TCP sender calculates CWND - varies in size much more quickly than does the advertised window, because it was designed to react to congestion.

The TCP sender always uses the lower of the two windows to determine how much data it can send before requiring an acknowledgment, by setting the value x into the Window field of the TCP header.

1) TCP sender fails to receive an ack in time, signifying a possible lost packet.
2) TCP sender sets CWND to the size of a single segment.
3) Slow start threshold SSTRESH is set to 50% of CWND value before lost segment.
4) Slow start governs how fast CWND grows until it reaches value of SSTRESH.
5) After CWND > SSTRESH congestion avoidance governs how fast CWND grows.

Slow start increases CWND by the MSS for every packet for which it receives an ack. CWND grows at an exponential rate during slow start. Congestion avoidance uses allows CWND to grow slower at a linear rate.

**Flow-based MQC**

- random-detect (dscp <dscp> | precedence <prec>) <min> <max> <mpd>

- random-detect exponential-weighting-constant <val>

**Legacy**

Can be configured only on main interfaces. Sets FIFO on interface

- random-detect - enable RED
- random-detect (dscp-based | prec-based)
- random-detect (dscp <dscp> | precedence <prec>) <min> <max> <mpd>
- random-detect exponential-weighting-constant <val>
- random-detect flow count <flows>
- random-detect flow average-depth-factor <#>

**Average queue size for a flow is a FIFO queue divided by number of flows which are identified by a hash.**

For each flow a flow depth is compared with scaled average queue size. If depth <= Average * Scale the flow is not randomly dropped

- random-detect
- random-detect MQC

**MPD**

Mark probability denominator

MPD=5 => (1/MPD) * 100% => 1/5 * 100% = 20%

One out of 5 packets is dropped during congestion

**REX**

WRED still randomly picks the packet, but instead of discarding, it marks a couple of bits in the packet header, and forwards the packet. Marking these bits begins a process which causes the sender to reduce CWND by 50%.

1) Both TCP endpoints agree that they can support ECN by setting ECN bits to either 01 or 10. If TCP sender does not support ECN, the bits should be set to 00. If ECN = 00 packet is discarded
2) Router checks the packet’s ECN bits, and sets the bits to 11 and forwards packet instead of discarding it.
3) TCP receiver notices ECN = 11 and sets Explicit Congestion Experienced (ECE) flag in the next TCP segment it sends back to the TCP sender.
4) TCP sender receives segment with ECE flag set and reduces CWND by half. TCP sender reduces CWND by half.
5) TCP sender sets Congestion Window Reduced (CWR) flag in next segment to inform receiver it slowed down.

**TCP behaviour**

The receiver grants the sender the right to send x bytes of data before requiring an acknowledgment, by setting the value x into the Window field of the TCP header.

The second window used by TCP is called the congestion window (CWND). TCP sender calculates CWND - varies in size much more quickly than does the advertised window, because it was designed to react to congestion.

The TCP sender always uses the lower of the two windows to determine how much data it can send before requiring an acknowledgment, by setting the value x into the Window field of the TCP header.

1) TCP sender fails to receive an ack in time, signifying a possible lost packet.
2) TCP sender sets CWND to the size of a single segment.
3) Slow start threshold SSTRESH is set to 50% of CWND value before lost segment.
4) Slow start governs how fast CWND grows until it reaches value of SSTRESH.
5) After CWND > SSTRESH congestion avoidance governs how fast CWND grows.

Slow start increases CWND by the MSS for every packet for which it receives an ack. CWND grows at an exponential rate during slow start. Congestion avoidance uses allows CWND to grow slower at a linear rate.

**Flow-based MQC**

- random-detect (dscp <dscp> | precedence <prec>) <min> <max> <mpd>

- random-detect exponential-weighting-constant <val>

**Legacy**

Can be configured only on main interfaces. Sets FIFO on interface

- random-detect - enable RED
- random-detect (dscp-based | prec-based)
- random-detect (dscp <dscp> | precedence <prec>) <min> <max> <mpd>
- random-detect exponential-weighting-constant <val>
- random-detect flow count <flows>
- random-detect flow average-depth-factor <#>

**Average queue size for a flow is a FIFO queue divided by number of flows which are identified by a hash.**

For each flow a flow depth is compared with scaled average queue size. If depth <= Average * Scale the flow is not randomly dropped

- random-detect
- random-detect MQC

**MPD**

Mark probability denominator

MPD=5 => (1/MPD) * 100% => 1/5 * 100% = 20%

One out of 5 packets is dropped during congestion

**REX**

WRED still randomly picks the packet, but instead of discarding, it marks a couple of bits in the packet header, and forwards the packet. Marking these bits begins a process which causes the sender to reduce CWND by 50%.

1) Both TCP endpoints agree that they can support ECN by setting ECN bits to either 01 or 10. If TCP sender does not support ECN, the bits should be set to 00. If ECN = 00 packet is discarded
2) Router checks the packet’s ECN bits, and sets the bits to 11 and forwards packet instead of discarding it.
3) TCP receiver notices ECN = 11 and sets Explicit Congestion Experienced (ECE) flag in the next TCP segment it sends back to the TCP sender.
4) TCP sender receives segment with ECE flag set and reduces CWND by half. TCP sender reduces CWND by half.
5) TCP sender sets Congestion Window Reduced (CWR) flag in next segment to inform receiver it slowed down.
Traffic metering is based on token bucket concept.

Shaping does not count TCP/IP headers and works only in outbound direction.

CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, match fr-dlci can be used.

CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, match fr-dlci can be used.

ISP usually polices input rate, and the customer usually shapes at the same rate to avoid tail dropping on ISP side.

Bc bits per Tc is the same ratio as CIR/sec but in smaller units (bursts)

During congestion adaptive shaping can drop traffic to minimum rate defined by MinCIR (50% of CIR by default)

FRTS + CBWFQ/LLQ

All classes within CBWFQ are processed by the scheduler, and then all outgoing packets are shaped. Bandwidth available for CBWFQ is a value defined as an average shape rate.

CBWFQ is used on a physical interface or subinterfaces.

CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, match fr-dlci can be used.

CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, match fr-dlci can be used.

CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, match fr-dlci can be used.

CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, match fr-dlci can be used.

CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, match fr-dlci can be used.

CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, match fr-dlci can be used.
CB policing replenishes tokens in the bucket in response to a packet arriving at the policing function, as opposed to using a regular time interval (Tc). Every time a packet is policed, CB policing puts some tokens back into the Bucket. The number of tokens placed into the Bucket is calculated as follows:

\[
\text{Tokens placed} = \frac{(\text{Current packet arrival time} - \text{Previous packet arrival time}) \times \text{Police rate}}{8}
\]

Policing does not use Tc but tokens. Number of tokens available is counted as:

\[
\text{Tokens available} = \frac{\text{Packet B bytes}}{4000 \text{ bytes} / 1000 \text{ bytes per 1ms}}
\]

Policing starts with credit 1000, and resets to this value every 1 sec if no traffic appears, otherwise 32000 would be collected after 1 sec (4 B/1ms).

**Concept**

Tokens are replenished at policing rate (CIR)

**Tokens**

- **Conform**: Action... exceed-action --- violate-action set-dscp-transmit 0 violate-action set-frde-transmit

**Ex. 128k rate – if 1sec elapsed between packets, CB will add 16000 tokens. If 0.1sec elapsed, CB will add 0.1sec’s worth of tokens 1600 tokens.**

**Tokens added at CIR Rate**

- Default for single-bucket: Bc = CIR/32 or 1500, whichever is larger, Be = Bc
- Default for dual-bucket: Bc = CIR/32, Be = Bc

**Tokens added at PIR Rate**

- police 32000 1000 conform-action... exceed-action... violate-action... set-dscp-transmit 0 violate-action... set-frde-transmit 0

**Tokens**

- **Be bucket**: Allows bursts until Be empties
- **Bc**: How fast tokens are replenished within 1 sec
- **Bc = (CIR/8)*(Tc/1000)**
- **Bc** and **Be** are not cumulative

**Two buckets; Three actions:** Conform, Exceed, Violate

**N N**

- B tokens from Be
- Violate (drop or remark)

**Two-rate**

**Conform**

- B tokens from Bc
- Conform

**Single-rate**

- B tokens from Bc
- Conform

**Two-color**

- B tokens from Bc?
- Violate (drop or remark)

**Three-color**

- B tokens from Bc?
- Conform
QoS uses mapping tables to derive internal DSCP from received CoS or IP prec. These maps include the CoS-to-DSCP map and the IP-precedence-to-DSCP map during policing, QoS can assign another DSCP value to an IP or non-IP packet (if the packet is out of profile and the policer specifies a marked down DSCP value). This configurable map is called the policed-DSCP map.

Before traffic reaches scheduling stage, QoS uses the DSCP-to-CoS map to derive CoS value from internal DSCP. Through CoS-to-egress-queue map, the CoS select one of the four egress queues for output processing.

The default policed-DSCP map is a null map, which maps an incoming DSCP value to the same DSCP value.

If the two domains have different DSCP definitions between them, use the DSCP-to-DSCP-mutation map to translate a set of DSCP values to match the definition of the other domain.

If QoS uses mapping tables to derive internal DSCP from received CoS or IP prec, these maps include the CoS-to-DSCP map and the IP-precedence-to-DSCP map.

Switch port trust state

Router

Auto QoS

Switch

Existing QoS configurations are overridden when Auto QoS is configured on port.

auto qos voip trust

The switch trusts CoS for switched ports or DSCP for routed ports.

auto qos voip cisco-phone

IF IP Phone is detected using CDP, port trusts CoS. If phone is not present, marking is reset to 0. Ingress and egress queues are configured.

auto qos voip cisco-softphone

Switch uses policing. packets do not have a DSCP value of 24, 26, 46 or is out of profile, the switch changes the DSCP value to 0.

auto qos trust dscp

If switch trusts IPPrec or DSCP and non-IP packet arrives then if COS field is present (trunk) then proper mapping is used to derive internal DSCP, but if COS is not present, the default COS, assigned statically is used. Bonus will not remark DSCP, but will remark the COS field based on the DSCP-to-COS map.

auto qos trust cos

If switch trusts COS then mapping is used for IP and non-IP packets on trunk. Switch will not remark COS, but will remark the DSCP field based on the DSCP-to-COS map.

auto qos cos <value>

Configures appropriate templates for voip traffic. FR-to-ATM interworking can be configured. Discovery phase is not required.

auto qos voip cisco-phone

Configures appropriate templates for voip traffic. FR-to-ATM interworking can be configured. Discovery phase is not required.

Auto discovery qos [trust]

Start the Auto-Discovery (data collection) phase, using NBAR to perform statistical analysis on the network traffic. Trust uses DSCP-to-IP to build class-maps.

Conditional marking. Enabled when switch detects IP Phone using CDPv2 auto qos voip [trust] (fr-aim)

If configured on FR links below 78K (broadband) MLPPP over FR (MLPPPFR) is configured automatically. Fragmentation is configured using a delay of 10 milliseconds (ms) and a minimum fragment size of 60 bytes.

IF auto discovery qos [trust]

Start the Auto-Discovery (data collection) phase, using NBAR to perform statistical analysis on the network traffic. Trust uses DSCP-to-IP to build class-maps.

IF auto qos

Generates templates based on data collection phase and installs them on interface. Discovery phase is required. Command is rejected without discovery process.

Network processor (NPU)

Generates templates based on data collection phase and installs them on interface. Discovery phase is required. Command is rejected without discovery process.

IP Precedence-mutation map is not a map, which maps an incoming DSCP value to the same DSCP value.

If switch trusts IPPrec or DSCP and non-IP packet arrives then if COS field is present (trunk) then proper mapping is used to derive internal DSCP, but if COS is not present, the default COS, assigned statically is used. Bonus will not remark DSCP, but will remark the COS field based on the DSCP-to-COS map.

IP Precedence-to-DSCP

Map IP precedence values in incoming packets to a DSCP value that QoS uses internally to represent the priority of the traffic.

Rewrite

Rewrite

Rewrite

Rewrite

Rewrite
VLAN based

Ingress Queue

Egress queue

3560 QoS

inter-VLAN service-policy in <policy-map>

## VLAN based

### Ingress Queue

**Scheduler - Shared Round Robin**

- **1. Define threshold levels**
  - You can prioritize traffic by placing packets with particular DSCPs or CoSs into certain queues and adjusting the queue thresholds so that packets with lower priorities are dropped first. Threshold 1 is always 100% (non-modifiable).
  
  - **mls qos srr-queue input threshold <Q1/2> <t1 %> <t2 %>
  
  - **2. Assign COS/DSCP to thresholds**
  - Using the following commands,
  
  ```
  mls qos srr-queue input dscp-map queue <Q1/2> threshold <T1/2/3> <dscp1-8>
  mls qos srr-queue input cos-map queue <Q1/2> threshold <T1/2/3> <cos1-8>
  ```

  - **3. Allocate memory buffers**
  - This command assigns the ingress buffer space to each queue and determines how much data can be buffered before packets are dropped.
  
  ```
  mls qos srr-queue input buffers <Q1%> <Q2%>
  ```

  - **4. Define bandwidth**
  - Bandwidth allocation control how much data can be buffered before packets are dropped.
  
  ```
  mls qos srr-queue input bandwidth <Q1 weight> <Q2 weight>
  ```

- **5. Define priority**
  - By default, 10% of Q2 is for priority traffic. Only one (overwrite) queue can have priority.
  
  ```
  mls qos srr-queue input priority-queue <Q1/2> bandwidth <% of interface>
  ```

### Egress Queue

**Marker**

- **1. Define threshold levels**
  - You can prioritize traffic by placing packets with particular DSCPs into certain queues and adjusting the queue thresholds so that packets with lower priorities are dropped first. Threshold 1 is always 100% (non-modifiable).
  
  - **mls qos srr-queue output threshold <Q1/2> <t1 %> <t2 %>
  
  - **2. Assign COS/DSCP to thresholds**
  - Using the following commands,
  
  ```
  mls qos srr-queue output dscp-map queue <Q1/2/3/4> threshold <T1/2/3/4> <dscp1-8>
  mls qos srr-queue output cos-map queue <Q1/2/3/4> threshold <T1/2/3/4> <cos1-8>
  ```

- **3. Allocate memory buffers**
  - All buffers must sum up to 100%.
  
  ```
  mls qos queue-set output <Set1/2> threshold <Q1/2/3/4> <T1> <T2> <Resv> <Max>
  ```

- **4. Define bandwidth**
  - Bandwidth allocation control how much data can be buffered before packets are dropped.
  
  ```
  mls qos srr-queue output bandwidth <Q1 weight> <Q2 weight>
  ```

### Aggregated policer

Aggregated policer is not working on 3560. To apply per-port per-vlan policer, nested policy can be applied with classes matching input interface.

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Q1</td>
</tr>
<tr>
<td>Q2</td>
<td>Q2</td>
</tr>
<tr>
<td>Q3</td>
<td>Q3</td>
</tr>
<tr>
<td>Q4</td>
<td>Q4</td>
</tr>
<tr>
<td>FE/GE</td>
<td>FE/GE</td>
</tr>
</tbody>
</table>

**QoS**

- **Shape**
  - Rate-limits queue up to queue bandwidth, even if other queues are empty. Weights in inverse ratio: 8 means 1/8 of BW.
  
  ```
  mls qos srr-queue bandwidth shape 8 0 0 0
  ```

- **Priority queue**
  - Policed up to 1/4th of BW. Used to define PQ
  
  ```
  mls qos priority-queue out
  ```

**Memory buffers**

- **Mark queue**
  - Set by default applied to all interfaces
  
  ```
  mls qos queue-set output <Set1/2> threshold <Q1/2/3/4> <T1> <T2> <Resv> <Max>
  ```

**Remaining BW**

Memory buffers

- **Intf BW**
  - Shared among other queues (Vlan ignored in ration calculations)

```
mls qos srr-queue bandwidth share <e1> <e2> <e3> <e4>
```
QoS
Ingress Queue
Egress queue

1. Configuring Minimum-Reserve Levels on FE ports
   wrr-queue min-reserve <queue-id> <MRL level>
   (G) mls qos min-reserve <level> <packets>
   There are 8 possible levels. By default, queue 1 selects level 1, queue 2 selects level 2, queue 3 selects level 3, and queue 4 selects level 4.

2. Mapping CoS Values to Select Egress Queues
   wr-queue cos-map <queue-id> <cos1> ... <cos8>
   COS/DSCP
   Remaining BW is shared among other queues (W4 is ignored in ration calculations)
   wr-queue bandwidth <queue> <W1> <W2> <W3> <W4>
   Priority queue (IF) priority-queue out
   tail-drop thresholds
   wr-queue threshold <queue> <T1> <T2>
   WRED thresholds
   wr-queue random-detect max-threshold <queue> <T1> <T2>

3. Allocating Bandwidth among Egress Queues
   wr-queue bandwidth <W1> <W2> <W3> <W4>
   Ratio of weights is the ratio of frequency in which WRR scheduler dequeues packets from each queue
   Each Q has 2 thresholds defined as % of Q len. Linear drop between T1 and T2 from 0 to 100%

4. Queues with classification based on COS (Q4 can be PQ)
   show mls qos interface <IF> queueing
   mls qos aggregate-police <name> <rate-bps> <burst-byte> exceed-action {drop | policed-dscp-transmit}
   class <name>
   policy aggregate <name>
   mls qos aggregate-police <name> <rate-bps> <burst-byte> exceed-action {drop | policed-dscp-transmit}
   class <name>
   policy aggregate <name>

5. Memory buffers
   wr-queue min-reserve <queue-id> <MRL level>
   wr-queue min-reserve <queue-id> <level> <packets>
   wr-queue min-reserve <queue-id> <level> <packets>
   wr-queue min-reserve <queue-id> <level> <packets>

6. Output
   wr-queue bandwidth <W1> <W2> <W3> <W4>
   Ratio of weights is the ratio of frequency in which WRR scheduler dequeues packets from each queue
   Each Q has 2 thresholds defined as % of Q len. Linear drop between T1 and T2 from 0 to 100%

7. Per-port Per-VLAN
   class-map match-any COMMON
   match ip dscp 24
   match ip address 100
   class match vlan 10 20-30 40
   match class-map COMMON

8. You cannot configure both port-based classification and VLAN-based classification at the same time. Hierarchical class-maps are required.
   Within a policy map, when you use the match vlan <vlan-list> command, all other class maps must use the match vlan <vlan-list> command.

9. By default all 64 DSCPs are mapped to T1
   wrr-queue dscp-map <threshold> <dscp1-8>

10. 1x FIFO; 8 policers per FE, 128 policers per GE
    Per-port Per-VLAN
    You cannot configure both port-based classification and VLAN-based classification at the same time. Hierarchical class-maps are required.

11. 1x FIFO: 8 policers per FE, 128 policers per GE
    Ingress Queue
    Egress queue

12. You can create a policer that is shared by multiple traffic classes within the same policy map. However, you cannot use the aggregate policer across different policy maps or interfaces.
    mls qos aggregate-police <name> <rate-bps> <burst-byte> exceed-action {drop | policed-dscp-transmit}
    class <name>
    policy aggregate <name>
    mls qos aggregate-police <name> <rate-bps> <burst-byte> exceed-action {drop | policed-dscp-transmit}
    class <name>
    policy aggregate <name>
Compression

The 40 bytes of the IP/UDP/RTP headers compress to between 2 and 4 bytes if no parameters are used, both IP and RTP headers are enabled. Payload

Stacker
CPU-intensive

HDLC

Frame-Relay
P-2-P

PPP
Memory-intensive

IP compress stac

((header ip [tcp |rtp])

if no parameters are used, both IP and RTP headers are enabled

Protocol

Class (class) compression [header ip [tcp |rtp]]

Stacker
CPU-intensive

HDLC

Frame-Relay
P-2-P

PPP
Memory-intensive

IP compress stac

Payload

The only method available on CBWFQ

The 40 bytes of the IP/UDP/RTP headers compress to between 2 and 4 bytes

Legacy

(IF) compress stac
cisco payload-compression FRF9 stac

(IF) compress stac

CISCO: broadcast tcp payload-compression packet-by-packet

CISCO: broadcast cisco payload-compression packet-by-packet

IETF: broadcast tcp payload-compression [passive]

Passive means the compression kicks in if the other end requests it by sending compressed header

(IF) ip {tcp | rtp} compression-connections <#>

Connections are unidirectional, so twice the required numbers have to be specified

Frame-relay per-VC p2p

(IF) frame-relay ip tcp header-compression [passive | active]

– enable compression for all VCs

(IF) frame-relay map ip <ip> <dlci> broadcast cisco payload-compression packet-by-packet

(IF) frame-relay map ip <ip> <dlci> broadcast tcp header-compression [passive]

(IF) frame-relay ip rtp header-compression

(IF) frame-relay map ip <ip> <dlci> nocompress

– disable compression for particular VC

CISCO:

(IF) frame-relay payload-compression packet-by-packet

IETF:

(IF) frame-relay map <ip> <dlci> broadcast ietf payload-compression FRF9 stac

(IF) frame-relay payload-compression frf9 stac

(IETF) frame-relay payload-compression FRF9 stac

(IF) frame-relay payload-compression packet-by-packet

PPP

HDLC

Frame-Relay
P-2-P

PPP
Memory-intensive

IP compress stac

LFI

Multilink is configured on a single physical interface

(IF) ppp multilink fragment-delay <msec>

(IF) ppp multilink interleave

Serialization delay becomes less than 10 ms for 1500-byte packets at link speeds greater than 768 kbps. Cisco recommends that LFI be considered on links with a 768-kbps clock rate and below

Dual FIFO queues created by FRF 12 create a high-priority queue

FRF 12 does not set maximum delay, as class MLP/LFI. Fragment size is configured directly

IF: 128 kbps additional 2 bytes of overhead are needed to manage the fragments

Fragment size = Max-delay * bandwidth (physical rate)

FRF 12 is used for in- and fragmentation is used

show frame-relay fragment <dlci>

16 configurable static round-robin queues. Default queue is 1

Queue 0 is a priority-like system queue served always first. Only L2 keepalives fall in there by default. Routing protocols should be assigned manually

Whole packet is always sent. If byte-count is 1501, and there are two 1500 byte packets, they will be both sent. No deficit schema

queue-list <nr> protocol ip <queue>

group-list <nr> default <queue>

queue-list <nr> queue-limit <high> <medium> <normal> <low> (# of packets)

queue-list <nr> queue-byte-count <bytes> (1500 bytes is default)

queue-list <nr> lowest-custom <queue>

Priority queue (served after system queue is emptied). Voice RTP can be assigned to that queue. This queue is not limited, so can starve other queues

(IF) custom-queue-list <nr>

Priority queuing

4 static queues: high, medium, normal, low

Every better queue is emptied before any other queue is emptied. Better queues are checked after each consecutive queue was served. Semiround-robin multid-queue

priority-list <nr> protocol {ip | http | ...} {high | medium | normal | low}

priority-list <nr> queue-limit <high> <medium> <normal> <low> (# of packets)

(IF) priority-group <nr>

Routing protocols are automatically prioritized. ARP goes to default queue

Legacy queueing mechanisms take L2 header into consideration

Legacy queuing
RSVP

**Operation**
- Sender sends a special RSVP packet called path messages to the network (contains Tspec)
- Path message flows through the network, along the normal routed path of data from the sender to the receiver. The direction of the message is downstream.
- The path messages are propagated from the source to the destination on a periodic basis (by default every 30 sec.). The reservation is active as long as messages are propagated.
- When an RSVP enabled router receives the path message, it keeps a record of the information contained in the message, this information contains: From, To, Previous hop, Requested bandwidth. PATH message does not reserve any resources.
- Once the receiver receives the path message, the receiver inspects the path message and uses the information in the path message to formulate an RSVP reservation request to the network, this message is called a Reservation message.
- When a router receives a Reservation Message it either accepts or rejects the Reservation message based on the available resources. RSVP message contains two structures: flowspec and filterspec.
- Once the Reservation message gets to the sender, it knows that the received QOS is in place and starts the transmission.

**Configuration**
- `ip rsvp bandwidth <total bw kbps> <single flow kbps>`
  - By default 75% of BW can be reserved.
- If RSVP BW is configured on sub intf it must be also configured on main interface as a sum of all sub intf BW values.
- Fair-queueing is required. FRTS classifier directing traffic to LLQ, so it must be added to the LL queue (frame-relay fair-queue)
- RSVP BW is subtracted from interface bandwidth available for CBWFQ.
- Proxy - if connected client is not RSVP-aware
  - `ip rsvp sender ...`
  - `ip rsvp reservation ...`
- LLQ - priority queue which should be used by LLQ
  - RSVP classifier directs flows matching reservation (flowspec) to CBWFQ LLQ. However, exceeding flows are not policed, although they use LLQ, but are remarked as best-effort
  - LLQ (priority queue) is not required in CBWFQ
  - `ip rsvp pq-profile <max-rate> <max-burst> <peak-to-avg ratio in %>`
- `ip rsvp sender-host <rcv IP> <snd IP> [tcp | udp] [sg dat port] <src port> <session bw kbps> <burst kbps>`
  - RSVP PATH signaling can be tested with this command
- `ip rsvp reservation-host <rcv IP> <snd IP> [tcp | udp] [sg dat port] <src port> [FF | SE | WF] <session bw kbps> <burst kbps>`
  - RSVP RESV signaling can be tested with (FF – fixed filter for single reservation, SE – shared explicit with limited scope, WF – wildcard filter with unlimited scope)

**Features**
- Flows are unidirectional, so each side has to request own RSVP path
- Traffic engineering is treated as a bottleneck
- RSVP reservations take precedence over user-defined classes in CBWFQ
- Poorly scalable – each flow requires own reservation. Used mainly for MPLS Traffic Engineering

**RESV**
- Flowspec
  - Contains sources which may use reservation installed by the receiver
  - SF – shared explicit – multiple, but explicitly defined sources can use the reservation (receiver specifies sources' IPs)
  - WF – wildcard filter – any sender can use the reservation
  - `ip rsvp sender-host <rcv IP> <snd IP> [tcp | udp] [sg dat port] <src port> <session bw kbps> <burst kbps>`
- Reservation specification (class of service requested)
- Traffic specification (parameters for traffic metering – Avg rate and burst)
- `ip rsvp bandwidth <total bw kbps> <single flow kbps>`
  - By default 75% of BW can be reserved.
- If RSVP BW is configured on sub intf it must be also configured on main interface as a sum of all sub intf BW values.
- Fair-queueing is required. FRTS classifier directing traffic to LLQ, so it must be added to the LL queue (frame-relay fair-queue)
- RSVP BW is subtracted from interface bandwidth available for CBWFQ.
- Proxy - if connected client is not RSVP-aware
  - `ip rsvp sender ...`
  - `ip rsvp reservation ...`
- LLQ - priority queue which should be used by LLQ
  - RSVP classifier directs flows matching reservation (flowspec) to CBWFQ LLQ. However, exceeding flows are not policed, although they use LLQ, but are remarked as best-effort
  - LLQ (priority queue) is not required in CBWFQ
  - `ip rsvp pq-profile <max-rate> <max-burst> <peak-to-avg ratio in %>`
- `ip rsvp sender-host <rcv IP> <snd IP> [tcp | udp] [sg dat port] <src port> <session bw kbps> <burst kbps>`
  - RSVP PATH signaling can be tested with this command
- `ip rsvp reservation-host <rcv IP> <snd IP> [tcp | udp] [sg dat port] <src port> [FF | SE | WF] <session bw kbps> <burst kbps>`
  - RSVP RESV signaling can be tested with (FF – fixed filter for single reservation, SE – shared explicit with limited scope, WF – wildcard filter with unlimited scope)
- `ip rsvp sender ...`
- `ip rsvp reservation ...`
- `ip rsvp installed [detail]`
Packets initiated by a router are not matched by outbound ACL or any inspection !!!

CBAC

Lock-and-Key (dynamic) ACL

Reflexive ACL

ACL

L3 Security

TCP intercept

Router replies to TCP Syn instead of forwarding it. Then, if TCP handshake is successful it establishes session with server and binds both connections

In watch mode, connection requests are allowed to pass but are watched until established. If they fail to become established within 30 sec IOS sends RST to server to clear up to state.

ACL can be applied as inbound to switch ports (L3 ports support L3 and L2 ACLs, and L2 ports support L2 ACLs only), but for outbound L2 only needs to be used

ACL permits tcp any any {match-all | match-any} +ack +syn -urg -psh ...

ACL supports only numbered ACLs

Only drop and police actions are available

The packet must be received at an interface that has the best return path (route) to the packet source. Reverse lookup in the CEF table is performed

Unicast RPF is an input function and is applied only on the input interface

Unicast RPF will allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) functions work properly

If an ACL is specified in the command, then when (and only when) a packet fails the Unicast RPF check, the ACL is checked to see if the packet should be dropped (using a deny statement in the ACL) or forwarded (using a permit statement in the ACL).

MOC supports only numbered ACLs

Only drop and police actions are available

Control-plane

service-policy (input | output) <name>

By Krzysztof Zaleski, CCIE #24081. This Booklet is available for free and can be freely distributed in a form as is. Selling is prohibited.
A zone is a group of interfaces that have similar functions or features from security perspective. Traffic between interfaces in the same zone is allowed. Requirements for policy enforcement:

- Traffic generated by the router or to the router is not a subject to any policy. A self-zone can be defined (no interfaces are assigned to it) to create policy for router traffic (not a traffic flowing through a router). Policing is not allowed in policies that are attached to zone-pairs involving a self-zone.
- When interface is added to a zone, all traffic is dropped. To allow traffic, a pair of zones must be defined with appropriate policy (pass, inspect).
- An interface cannot be part of a zone and legacy inspect policy at the same time.

ACLs applied to interfaces that are members of zones are processed before the policy is applied on the zone-pair.

- zone security <name> {source <zone-name> | self} destination <zone-name>
- service-policy type inspect <name>
- (IF) zone-member security <zone-name>
- (G) zone security <name> {source <zone-name>} destination <zone-name>
- policy-map type inspect <name>
- class type inspect <name>
- match access-group {<acl> | name <acl-name>}
- match protocol <name> [signature]

A Layer 7 policy map must be contained in a Layer 3 or Layer 4 policy map; it cannot be attached directly to a target.

FastTrack, eDonkey, eMule, H.323, HTTP, Kazaa, ICQ, MSN IM, POP3, SIP, SMTP, SunRPC, NBAR is not available for bridged packets (transparent firewall between bridged interfaces).

Protocol-specific parameter maps can be created only for Instant Messenger applications.

- parameter-map type protocol-info <name>
- match protocol <name> [signature]
- service-policy type inspect <name>
- policy-map type inspect <name> {source <zone-name> | self} destination <zone-name>
- service-policy type inspect <name>
- policy-map type inspect <name>
- class type inspect <name>
- match access-group {<acl> | name <acl-name>}
- match protocol <name> [signature]
- policy-map type inspect <name>
- class type inspect <name>
- match access-group {<acl> | name <acl-name>}
- match protocol <name> [signature]
- policy-map type inspect <name>
- class type inspect <name>
- match access-group {<acl> | name <acl-name>}
- match protocol <name> [signature]

NBAR is not available for bridged packets (transparent firewall between bridged interfaces).

Protocol-specific parameter maps can be created only for Instant Messenger applications.

- parameter-map type protocol-info <name>
- match protocol <name> [signature]
- service-policy type inspect <name>
- policy-map type inspect <name> {source <zone-name> | self} destination <zone-name>
- service-policy type inspect <name>
- policy-map type inspect <name>
- class type inspect <name>
- match access-group {<acl> | name <acl-name>}
- match protocol <name> [signature]
- policy-map type inspect <name>
- class type inspect <name>
- match access-group {<acl> | name <acl-name>}
- match protocol <name> [signature]

Default behaviour is to drop packets arriving out of order.

- Out-of-Order
  - tcp reassembly alarm (on | off)
  - tcp reassembly memory limit <limit>
  - tcp reassembly queue length <size>

- clear zone-pair inspect sessions
  - Changes to the parameter map are not reflected on connections already established through the firewall.
In-line intrusion detection sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures.

**Actions:** Send an alarm to a syslog server. Drop the packet. Reset the connection. Deny traffic from the source IP address of the attacker for a specified amount of time. Deny traffic on the connection for which the signature was seen for a specified amount of time.

A transparent Cisco IOS IPS device acts as a Layer 3 (only) IPS between bridged interfaces. A transparent IPS device supports an IPS for outgoing traffic.

If you want to configure transparent IPS, you must configure a bridge group before loading IPS onto a device.

**Features**

- IPS between interfaces.
- IPS configuration is always available on flash with IOS IPS. If neither file is specified, IOS uses internal built-in signatures.
- Signature version 4 and version 5 are backward compatible.
- Signature version 4 and version 5 are both built-in.
- Signature version 4 and version 5 are both stored in NVRAM.
- Signature version 4 and version 5 are both loaded into the IPS table.
- Signature version 4 and version 5 are both saved to the global configuration file.

**Signature Categories**

Signatures are pregrouped into hierarchical categories. Signature can belong to more than one category.

**Reporting**

- Reporting can be done using syslog or SDEE (Security Device Event Exchange).
- SDEE is an application-level protocol used to exchange IPS messages between IPS clients and IPS servers. It is always running but it does not receive and process events from IPS unless SDEE notification is enabled.
- To use SDEE, the HTTP server must be enabled.

**Tuning**

- Tuning parameters for the IPS feature can be configured using the CLI.
- Tuning parameters for the IPS feature can be configured using the GUI.
- Tuning parameters for the IPS feature can be configured using the REST API.

**Verify**

- Show IPS configuration
- Show IPS signature definition
- Show IPS signature-category
- Show IPS signature-action
- Show IPS signature-notification
- Show IPS signature-event-action
- Show IPS signature-event-action-rules
- Show IPS signature-event-action-rules
- Show IPS signature-event-action-rules
- Show IPS signature-event-action-rules
ip dhcp snooping

If aggregation switch with DHCP snooping receives option-82 from connected edge switch, the switch drops packets on untrusted interface. If received on trusted port, the aggregation switch cannot learn DHCP snooping bindings for connected devices and cannot build a complete DHCP snooping binding database.

ip dhcp snooping database <filesystem>

No ip dhcp relay information option

(ip) ip dhcp snooping trust

Enable ports with trusted devices (DHCP server)

(ip) ip dhcp snooping trust

Define restricted VLAN upon authentication failure. The user is not notified of the authentication failure.

mac-address-table static 0000.1111.1111 vlan <vlan> drop

Permit any any interfaces. Cat 3550 treats IPv6 as non-IP

mac-address-table static 0000.1111.1111 vlan <vlan> interface <if>

Port blocking

Prevent unknown unicast or multicast traffic from being forwarded from one port to another

(ip) switchport block {unicast | multicast}

Dot1x guest-vlan <vlan-id>

Dot1x port-control {auto | force-authorized | force-unauthorized}

EAP provides link layer security framework. It can run on any data link (802, PPP).

aaa authentication dot1x group

dot1x system auth-control

Enables dot1x (required)

EAPoL

Dot1x timeout reauth-period <sec>

Dot1x host-mode multi-host

Access all hosts connected to one port to use authentication performed only by one host

dot1x host-mode <mode>

Define restricted VLAN upon authentication failure. The user is not notified of the authentication failure.

dot1x auth-fail vlan <vlan>

The switch assigns a VLAN to a guest VLAN when it does not receive a response to EAPOL

dot1x auth-fail default-vlan <vlan-id>

The dot1x host-mode multi-host allows multiple hosts to authenticate using the same VLAN.

dot1x auth-fail default-vlan <vlan-id>

Mac address-table static 0000.1111.1111 vlan <vlan> interface <if>

Mac address-table static 0000.1111.1111 vlan <vlan> drop

MAC ACL

Filter only non-IP traffic per MAC address. Cat 3550 treats IPv6 as non-IP

mac-access-list extended <name>

deny any any

mac-access-list extended <name>

permit any any

Static MAC

Mac address-table static 0000.1111.1111 vlan <vlan> interface <if>

VLAN ACLs are inbound and they can conflict with other per-port filters

VLAN ACLs run in hardware. They must be re-applied if changed

VLAN access-map <name> keep (access-map is like route-map, many entries with different actions)

match ip [map | mac] address <aclname> action (drop | forward)

VLAN filter <name> vlan-list <vlans>

Expects that there is no exchange of unicast, broadcast, or multicast traffic between ports on the switch

Storm-control

All data traffic passing between protected ports must be forwarded through a Layer 3 device. ICMP redirects are automatically disabled on protected ports.

storm-control (broadcast | multicast | unicast) level pps <high> [burst <intv>]

storm-control action (shutdown | lookup)

Port security

When rate of mac traffic exceeds a threshold, all incoming traffic (broadcast, multicast, and unicast) is dropped. Only spanning-tree packets are forwarded. When burst and unicast thresholds are exceeded, traffic is blocked for only the type of traffic that exceeded the threshold.

switchport port-security maximum [ip vlan | access] 2

If HSRP is used, configure n+1 allowed MACs. Also, if IP phone is used, define at least 3 MACs

switchport port-security mac-address <MAC> [vlan | access | voice]

Static MAC

dot1x host-mode multi-host

Only active ports can have dot1x requests. Port MUST be in access mode. If the port is configured as a voice VLAN port, the port allows VoIP traffic before the client is successfully authenticated.

802.1x

dot1x mac-access-list <name>

mac-access-list extended <name>

deny any any

mac-access-list extended <name>

permit any any

Supplicant

AAA new-model

Enables AAA new-model and defines authentication method for dot1x requests

dot1x system auth-control

Enable dot1x (required)

EAPoL

Dot1x guest-vlan <vlan-id>

The switch assigns a VLAN to a guest VLAN when it does not receive a response to EAPOL

VLAN access-map <name> keep (access-map is like route-map, many entries with different actions)

match ip [map | mac] address <aclname> action (drop | forward)

VLAN filter <name> vlan-list <vlans>
IP Traffic Export

 allows users to configure their router to export IP packets that are received on multiple, simultaneous WAN or LAN interfaces. It is similar to SPAN on switches.

 By default, only incoming traffic is exported.

 ip traffic-export profile <profile-name> interface <intf> (outgoing interface)
 bidirectional
 mac-address <H.H.H> (destination host that is receiving the exported traffic)
 incoming (access-list <acl>) | sample one-in-every <packet-#> | sample one-in-every <packet-#> interface (interface name)
 ip traffic-export apply <profile-name>

 Other Security

 View authentication is performed by attribute "cli-view-name."
 enable view
 parser view <view-name> secret "<pass>"
 commands <parser-mode> (include | include-exclusive | exclude) [interface <intf> | <command>]

 lawful-intercept view restricts access to specified commands and configuration information.
 enable view
 lawful-intercept <domain-name> user <username> password <password>

 Lawful-intercept view

 Allows administrator to assign all users within configured CLI views to a superview instead of having to assign multiple CLI views to a group of users.

 enable view
 parser view <superview-name> supernode "<pass>"
 supernode secret "<pass>"
 view "<view-name>" (Adds a normal CLI view to a superview)

 SSH

 ip ssh version [1 | 2] Both SSH v1 and 2 are enabled by default. If any version is defined, only this version is supported.
 ip ssh transport (input) ssh client
 ip ssh port <#> rotary <group> Connect the port with rotary group, which is associated with group of lines.

 crypto key generate rsa If RSA key pair is generated then it automatically enables SSH. To use SSH2 the key must be at least 768 bits.
 crypto key generate rsa Delete the RSA key pair.

 Allow user to gather information about the traffic that is flowing to a host that is suspected of being under attack and to easily trace an attack to its entry point into the network.

 ip source-track <ip-address> destination address being attacked (configured on a router closest to tracked source).
 ip source-track address-limit <number>
 ip source-track syslog-interval <1-1440 min>
 show ip source-track [ip-address] [summary | cache]

 Role-based CLI

 View authentication is performed by attribute "cli-view-name."
 enable view
 parser view <view-name> secret "<pass>"
 commands <parser-mode> (include | include-exclusive | exclude) [interface <intf> | <command>]

 lawful-intercept view restricts access to specified commands and configuration information.
 enable view
 lawful-intercept <domain-name> user <username> password <password>

 Lawful-intercept view

 Allows administrator to assign all users within configured CLI views to a superview instead of having to assign multiple CLI views to a group of users.

 enable view
 parser view <superview-name> supernode "<pass>"
 supernode secret "<pass>"
 view "<view-name>" (Adds a normal CLI view to a superview)
IPv4 packet

MPLS packet

CEF

CN

L2 header

Forwarding Plane

Control Plane

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB

From all remote bindings the best one is chosen and placed in LFIB. RIB is checked for best path to a prefix, then LSR, which is the next hop for that prefix is selected as best source for label in LFIB.

For every LSR that has a local prefix, RIB is checked and results are cached. The best path is selected and placed in LFIB.

BGP (VPN)

RSVP (TE)

IPV4

Label exchange protocols are used to bind labels to FECs

IPv4 packet

MPLS packet

CEF

CN

From all entries in a routing table, changes are automatically reflected in FIB. Adjacency table where recursive next-hops are automatically and immediately resolved.

RIB

LFIB
LSP is unidirectional
Aggregation breaks LSP into separate LSPs. Connectivity may be maintained for plain IPv4, but VPN and TE may be broken.

**Concept**

**Distribution Modes**
- DOD - Downstream on Demand: Request binding for FEC from next-hop LSR (only one binding in LIB) - ATM interfaces
- UD - Unsolicited Downstream: LSR propagates local bindings to all neighbors even if label was not requested - Frame mode
- CR - Conservative: Bindings are removed from LIB after best next-hop is selected and placed in LIB. Only best binding is stored in LIB - less memory but slow convergence - default for ATM interfaces
- LLR - Liberal: Always faster convergence when label goes downstream next-hop is selected from LIB
- SDL - Split Downstream: Each LSR creates bindings for connected prefixes immediately, but for other prefixes only after it receives remote bindings from next-hop LSR. Default for ATM interfaces

**Retention Modes**
- Ordered: Each LSR creates bindings for connected prefixes immediately, but for other prefixes only after it receives remote bindings from next-hop LSR. Default for ATM interfaces
- Independent: Each LSR creates bindings for prefixes as soon as they are in routing table (forwarded and received from IGPs) may cause a packet drop if LSR starts labeling packets and the whole LSP is not set up yet
- Dynamic: Set to 255. Egress LSR does not copy label TTL into IP TTL core is hidden. One hop is shown with maximum delay.

**Control Modes**
- TTL: Propagation is enabled by default. If MPLS TTL is higher than IP TTL on egress router than IP TTL is overwritten with label TTL otherwise it is not (loop prevention)
- Loop: MPLS MTU can be set to 1500. MPLS MTU must be set properly on both sides of the link. If fragmentation is needed of labeled IPv4 packet, LSR pops whole label stack, fragments IP and pushes whole shim header with valid stack for outgoing interface. Non-IPv4 packets are dropped.

**Service Modes**
- PE - Provider Edge: MPLS network termination at PE or PE router in MPLS network. PE router uses the shim header between label and IP header. PE router does not know how to reach a sender (no VPN knowledge). Egress LSR responds by forwarding ICMP back to sender. Only IPv4 and IPv6 packets can use ICMP Time Exceeded. ATM packets are dropped, as they contain L2 header behind label.

**MPLS Labels**

**Label Stack**

<table>
<thead>
<tr>
<th>L2 header</th>
<th>TE label</th>
<th>LSP label</th>
<th>VPN label</th>
<th>IP Header</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>S=0</td>
<td>S=0</td>
<td>S=0</td>
<td>S=1</td>
<td>S=1</td>
<td>S=0</td>
</tr>
</tbody>
</table>

**Debug MPLS Packet**

S = bottom of the stack, S=0 = bottom label, next to IP header. MPLS MTU must be set properly on both sides of the link. If fragmentation is needed of labeled IPv4 packet, LSR pops whole label stack, fragments IP and pushes whole shim header with valid stack for outgoing interface. Non-IPv4 packets are dropped.

**IF mpls mtu 1500**

<table>
<thead>
<tr>
<th>IF mpls mtu 1512</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500</td>
</tr>
</tbody>
</table>

**IF mpls ip**

<table>
<thead>
<tr>
<th>IF mpls ip 1500</th>
</tr>
</thead>
<tbody>
<tr>
<td>1512</td>
</tr>
</tbody>
</table>

**IF mpls mtu 1508**

<table>
<thead>
<tr>
<th>IF mpls mtu 1508</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500</td>
</tr>
</tbody>
</table>

**IF mpls mtu 1512**

Defines how large a labeled packet can be. Recommended 1512 for 3 labels (baby giant).

**IF mpls ip**

When MPLS is enabled on LAN interface, MPLS MTU is automatically increased when labeled packet is sent. But on WAN interfaces MPLS MTU stays the same as IP MTU, so in fact IP MTU is decreased (fragmentation).

**IF mpls mtu 1508**

All devices along the L2 path must support baby giant frames.
LDP Link Hello – every 5 sec, holdtime is 15 sec. If routers advertise
`mpls ldp discovery hello interval <sec>`
`mpls ldp discovery hello holdtime <sec>`
Different holdtimes the lower one is used by both. Interval is not advertised.

LDP Hello – UDP/646 to 224.0.0.2 (all routers) – even after TCP session is established – to discover new neighbors

Keepalive timer is reset every time LDP packet or keepalive (60 sec) is received.

Default holdtime is 180 sec. Keepalive is automatically adjusted to 1/3 of holdtime.

LDP identifier is 6 byte (4 byte router identifier, 2 byte label space identifier). Highest IP on all loopback interface is used first or highest IP any other active IP interface.

If initialization message cannot negotiate parameters (incompatibility), session is re-established in
`mpls ldp backoff <initial> <max>`

If ID is changed all interfaces must be shut/no shut – clearing session does not work. If force is used, all sessions are automatically hard-resetted.

**Keepalive**

LDP relies on IGP and label TTL for loop prevention.

Multipath sessions can be established between the same LDP peer if interface information was used.

Because labels are announced in a form of (LDP ID, label) for certain prefix, router must have mappings for all neighbor's interface IPs (to find next-hops). The Address Message announces them (bound addresses).

**Label distribution**

To prevent loops, labels are distributed in a controlled way by IGP and LDP. Each node only has a local view of the global label space. LDP sessions can be established between the same LDP peer if interface information was used.

If no LDP session is established, LDP does not provide signaling like BGP.

**Label allocation**

LDP identifier is 6 byte (4 byte router identifier, 2 byte label space identifier). Highest IP on all loopback interface is used first or highest IP any other active IP interface.

**Authentication**

If none of global MD5 password options matches neighbor, last-resort password can be used (catch all)

**Timers**

`mpls ldp discovery hello interval <sec>`
`mpls ldp discovery hello holdtime <sec>`
LDP Link Hello – every 5 sec, holdtime is 15 sec. If routers advertise different holdtimes the lower one is used by both. Interval is not advertised.

**Label distribution control**

Label space: Per-interface (>0). Per-platform (0) – the same label can be used on any interface. Not secure as some router can use label not assigned to him). Requires only one session between LSFRs if multiple parallel links exist between them. Frame mode works only for frame-mode interfaces. For example advertise labels only for loopback IPs which are BGP next hop addresses. Conditional propagation is not only for local prefixes but also for advertised by peers, so ACL must match appropriate range.

**Neighbors**

If initialization message cannot negotiate parameters (incompatibility), session is re-established in
`mpls ldp backoff <initial> <max>`

**Layer 2 Sessions**

If session is established, LDP does not provide signaling like BGP.

**Frame mode**

Auto-configuration

When IGP is up but LDP session is down then LSR installs unlabelled route to destination and packet is forwarded in a native form. Can break VPN.

**IGP synchronization**

OSPF is not synchronized.

**Auto-configuration**

**Auto-configuration**

OSPF supports synchronization. It announces link with max cost until LDP session is up. Hop is also not send on link where LDP is down or until synchronization timer expires. However, OSPF adjacency is formed if LDP detects that this link is the only one to reach neighbor's LDP ID.

**Graceful restart**

Enable SIG/SIGF graceful restart capability for LDP. Must be enabled before session is established.

**Graceful restart**

Amount of time (default 120s) a router waits for LDP session to be reestablished

**Multi-hop Hello**

Instead of adding mpls ip on each interface, LDP can be enabled on interfaces where specific IGP is enabled, but LDP MUST be enabled globally (mpls ip). Currently only OSPF and ISIS is supported.

PLS can be enabled on all interfaces where OSPF runs or only for specific area.

**Route reflector**

If route reflector is configured, LDP must be enabled globally (mpls ip).

(mpls ldp ip autoconf [area 0|1])

Enable autoconfiguration on specific interface

**Auto-configuration**

If autoconf is enabled for IGP, MPLS can be disabled globally (no mpls ip) only if autoconf is removed first

**Route reflector**

If route reflector is configured, LDP must be enabled globally (mpls ip).

(mpls ldp ip autoconf [area 0|1])

Enable autoconfiguration on specific interface

**Auto-configuration**

If autoconf is enabled for IGP, MPLS can be disabled globally (no mpls ip) only if autoconf is removed first.
L3 VPN

**Concept**

VPN labels are exchanged between edge LSRs. They describe to which VRF packet will be sent when it reached egress LSR. Intermediate LSRs do not have information about VPN labels. They only use top label (LDP) to pass traffic.

**IPv4 Update for 10.0.10.0/24**

- Static, eBGP
- Next Hop: 150.1.1.2
- OSPF, EIGRP

**Peer-to-peer:** IPSec, GRE, L2F, L2TP, PPTP

- RIPv2, ISIS
- Legacy

**Overlay:** FR, ATM VCs. ISP provides L1/L2 (usually expensive), and does not participate in customer’s routing.

**MP-BGP (iBGP) – address-family vpnv4**

- Lo0:150.1.1.1
- Lo0:150.1.1.2

**Multiprotocol Capabilities**

- Multiprotocol capabilities are exchanged in Open message
- Each attribute has two identifying field: AFI (2 bytes) and SAFI (1 byte)

**Address Families**

- Address-family vpnv4
- BGP prefix and label exchange between PE LSRs

**Route Target**

- Native BGP sessions for IPv4
- Each route is received by egress LSR, route is added to VRF. If local RD is different than RD received from BGP, it is stripped and local RD is added

**Route Distinguisher**

- Route Target
- MP-BGP

**Display BGP sessions in all VRF and VPNv4 families**

- show ip bgp vpnv4 all summary
- show ip vrf [id]

**Export route map**

- Can add RT to selected routes. No other action is overwritten in the prefix, unless additive keyword is used in route-map

**Import and export the same RT. Actually it is a macro creating the above two entries (import and export)**

**Supported by only basic MPLS L3 VPNs (Inter-AS and CSC are not supported). It is configured per-AS**

**MP-BGP**

- Site A
- Site B
- Site C

**Convergence**

- bgp scan-time (import) <10 sec

By Krzysztof Zaleski, CCIE #24081. This booklet is available for free and can be freely distributed in a form as is. Selling is prohibited.
Only one process is allowed per router so address-family is used for each VRF. Globally defined AS is used ONLY for native IPv4. You MUST define AS for address-family even if it is the same as global AS.

Extended communities are used to describe the route.

Extended community Type 0x4301
- Site of Origin – used for loop prevention in dual-homed CE when race condition between EIGRP queries and BGP updates takes place. Attached to VPNv4 route as extended community. EIGRP carries SOO as separate TLV. SOO cannot be used as backup for partitioned CE. This solution is slower in convergence, but provides redundancy.

Extended community Type 0x8802
- Reliability + Hop count + BW

Extended community Type 0x8803
- Reserved + Load + MTU

Extended community Type 0x8805
- Remote protocol + Remote metric

Cost community

<table>
<thead>
<tr>
<th>Type</th>
<th>POI</th>
<th>ID</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>110</td>
<td>19</td>
<td>49</td>
</tr>
</tbody>
</table>

When routes are redistributed from EIGRP into MP-BGP, cost community (non-transitive) is added. It carries the composite EIGRP metric in addition to individual EIGRP attributes.

MP-BGP update is a last-in first-out basis. EIGRP update is overwritten by MP-BGP updates. Peers without any route advertised by MP-BGP are not reset. MP-BGP updates are compared based on ID. Earlier is better. In case of a tie, lower is better. EIGRP ID is a tiebreaker when costs are the same. Lower is better.

Internet access

Other solutions are separate PE-CE circuit for native internet access with full BGP feed (native IPv4 peerings), egress with Internet VRF or VRF-aware NAT.
Regardless of area number on both PEs, internal routes (LSA 1, 2 and 3) are carried as inter-area (LSA 3) routes, even though they are redistributed from MP-BGP to OSPF. External routes are still carried as LSA5. PE becomes ABR (not ASBR). MPLS becomes superbackbone.

Area 0 is required on PE only if there is more than one area in the same domain (customer vrf). There is no adjacency established, nor flooding over MPLS VPN superbackbone for customer sites, except when sham-links are used.

Information about route is propagated using extended community called RT (route type, different than route target), OSPF router ID (4 bytes), and OSPF domain (process number) ID (2 bytes)

\[ RT: \text{area 4Bytes}:\text{route type 1Byte}:\text{options 1Byte} \]

Area (originating) is in dotted decimal form. Set to 0.0.0.0 if route is external. Route type: 1 or 2 – intra-area, 3 – inter-area, 5 – external. 7 – external nssa, 129 – sham-link endpoints. If least significant bit in options field is set then route is Type 2.

\[ \text{RT:<area 4Bytes>:<route type 1Byte>:<options 1Byte>} \]

Domain ID is the second community carried via MP-BGP. By default it is the OSPF process ID. If domain is different on both PEs than internal (LSA 1, 2, and 3) routes become LSA 5 Type 2 when sent to the other PE and redistributed from MP-BGP into OSPF.

Cost from internal and external nodes is copied into MED. MED can be manipulated manually to influence path selection.

Intra-area route is preferred than inter-area. If backup link exists between sites it will be preferred no matter what cost inter-area routes have. Also OSPF has lower AD (110) than BGP (200). OSPF adjacency is established. LSAs are exchanged, but they are used only for path calculations. Forwarding is still done using MP-BGP

Although sham link floods LSA 1 and 2, those routes must still be advertised through MP-BGP so labels are properly propagated. Routes in OSPF database are now seen as intra-area, even though they are seen via superbackbone.

Two /32 loopbacks are required for each link, as a source and destination of sham link. They must belong to VRF, but MUST NOT be advertised through OSPF, only via MP-BGP

Cost should be set to lower value if it is preferred over backup link.

\[ (\text{BGP}) \text{ network } <\text{ip} > \text{mask } 255.255.255.255 \text{ since ip ospf sham-link} \]

OSPF capability rid:

Required on CE if VRF LSA is used. Down-bit will not be taken into consideration, otherwise blackholing may occur. If this capability is not supported, all PEs should be configured with different domain id, so routes are redistributed as a LSA, which does not fall under the loop-prevention scheme.