# CCIE
# Routing and Swithing
# Quick Review Kit

**By: Krzysztof Załęski**
**CCIE R&S #24081**

**ver. 20100507**

## 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.

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 cadidate'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 encouradged 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 autor of this Booklet takes no responsibility, nor liablity 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.

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**

**Switching**

**IP Services**

**Routing**

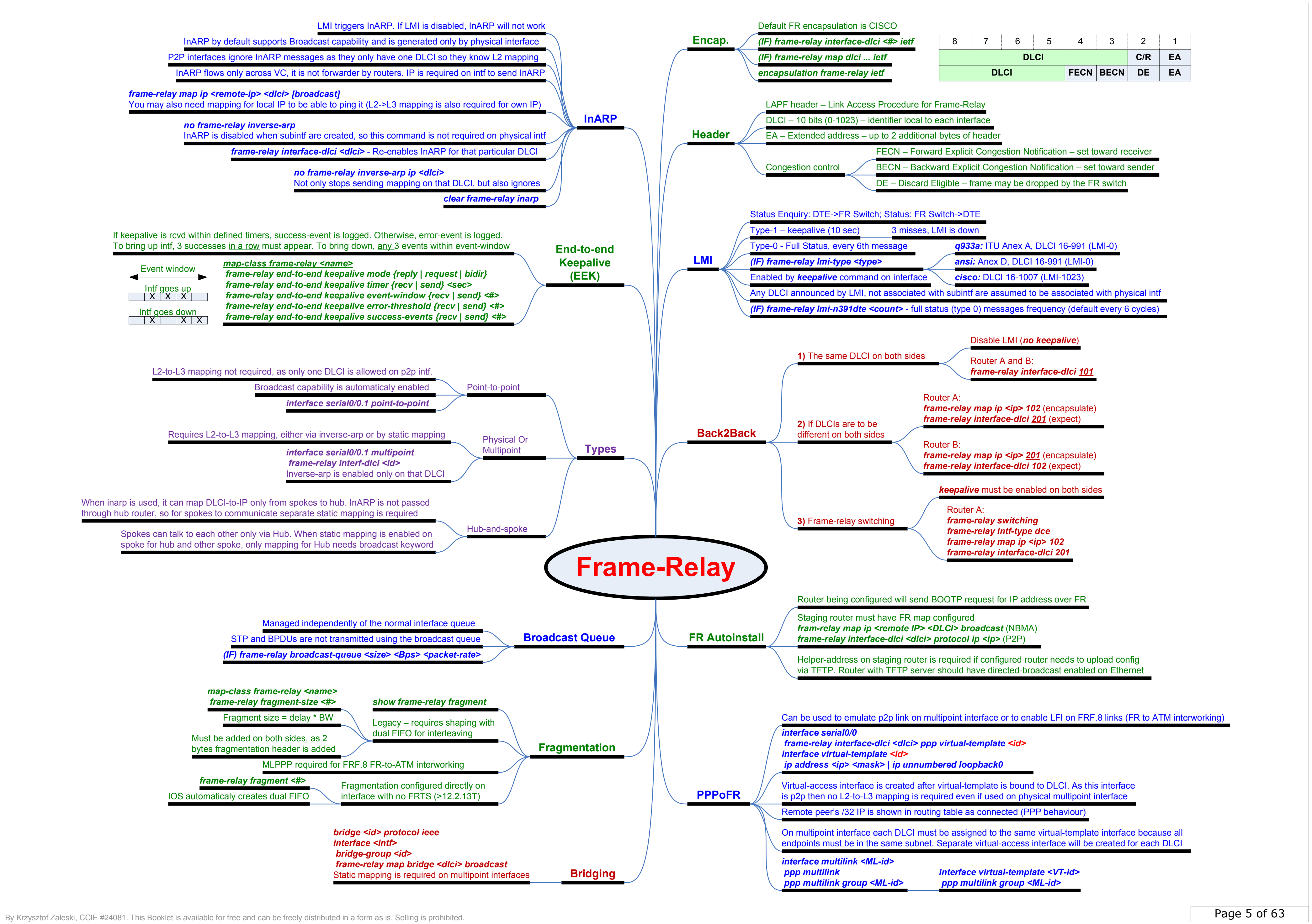
**IPv6**

**Multicast**

**Quality-of-Service**

**Security**

**MPLS**



# Frame-Relay

## InARP

- LMI triggers InARP. If LMI is disabled, InARP will not work
- InARP by default supports Broadcast capability and is generated only by physical interface
- P2P interfaces ignore InARP messages as they only have one DLCI so they know L2 mapping
- InARP flows only across VC, it is not forwarder by routers. IP is required on intf to send InARP
- ***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 own IP)
- ***no frame-relay inverse-arp***
  InARP is disabled when subintf are created, so this command is not required on physical intf
- ***frame-relay interface-dlci <dlci>*** - Re-enables InARP for that particular DLCI
- ***no frame-relay inverse-arp ip <dlci>***
  Not only stops sending mapping on that DLCI, but also ignores
- ***clear frame-relay inarp***

## End-to-end Keepalive (EEK)

- If keepalive is rcvd within defined timers, success-event is logged. Otherwise, error-event is logged.
  To bring up intf, 3 successes in a row must appear. To bring down, any 3 events within event-window

- ***map-class frame-relay <name>***
  ***frame-relay end-to-end keepalive mode {reply | request | bidir}***
  ***frame-relay end-to-end keepalive timer {recv | send} <sec>***
  ***frame-relay end-to-end keepalive event-window {recv | send} <#>***
  ***frame-relay end-to-end keepalive error-threshold {recv | send} <#>***
  ***frame-relay end-to-end keepalive success-events {recv | send} <#>***

## Types

- Point-to-point
  - L2-to-L3 mapping not required, as only one DLCI is allowed on p2p intf.
  - Broadcast capability is automaticaly enabled
  - ***interface serial0/0.1 point-to-point***
- Physical Or Multipoint
  - Requires L2-to-L3 mapping, either via inverse-arp or by static mapping
  - ***interface serial0/0.1 multipoint***
    ***frame-relay interf-dlci <id>***
    Inverse-arp is enabled only on that DLCI
- Hub-and-spoke
  - When inarp is used, it can map DLCI-to-IP only from spokes to hub. InARP is not passed through hub router, so for spokes to communicate separate static mapping is required
  - Spokes can talk to each other only via Hub. When static mapping is enabled on spoke for hub and other spoke, only mapping for Hub needs broadcast keyword

## Broadcast Queue

- Managed independently of the normal interface queue
- STP and BPDUs are not transmitted using the broadcast queue
- ***(IF) frame-relay broadcast-queue <size> <Bps> <packet-rate>***

## Fragmentation

- ***show frame-relay fragment***
- Legacy – requires shaping with dual FIFO for interleaving
  - ***map-class frame-relay <name>***
    ***frame-relay fragment-size <#>***
  - Fragment size = delay * BW
  - Must be added on both sides, as 2 bytes fragmentation header is added
- MLPPP required for FRF.8 FR-to-ATM interworking
- Fragmentation configured directly on interface with no FRTS (>12.2.13T)
  - ***frame-relay fragment <#>***
  - IOS automaticaly creates dual FIFO

## Bridging

- ***bridge <id> protocol ieee***
  ***interface <intf>***
  ***bridge-group <id>***
  ***frame-relay map bridge <dlci> broadcast***
  Static mapping is required on multipoint interfaces

## Encap.

- Default FR encapsulation is CISCO
- ***(IF) frame-relay interface-dlci <#> ietf***
- ***(IF) frame-relay map dlci ... ietf***
- ***encapsulation frame-relay ietf***

| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| DLCI | | | | | | C/R | EA |
| DLCI | | | | FECN | BECN | DE | EA |

## 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
- Congestion control
  - FECN – Forward Explicit Congestion Notification – set toward receiver
  - BECN – Backward Explicit Congestion Notification – set toward sender
  - DE – Discard Eligible – frame may be dropped by the FR switch

## LMI

- Status Enquiry: DTE->FR Switch; Status: FR Switch->DTE
- Type-1 – keepalive (10 sec) — 3 misses, LMI is down
- Type-0 - Full Status, every 6th message
- ***(IF) frame-relay lmi-type <type>***
  - ***q933a:*** ITU Anex A, DLCI 16-991 (LMI-0)
  - ***ansi:*** Anex D, DLCI 16-991 (LMI-0)
  - ***cisco:*** DLCI 16-1007 (LMI-1023)
- Enabled by ***keepalive*** command on interface
- Any DLCI announced by LMI, not associated with subintf are assumed to be associated with physical intf
- ***(IF) frame-relay lmi-n391dte <count>*** - full status (type 0) messages frequency (default every 6 cycles)

## Back2Back

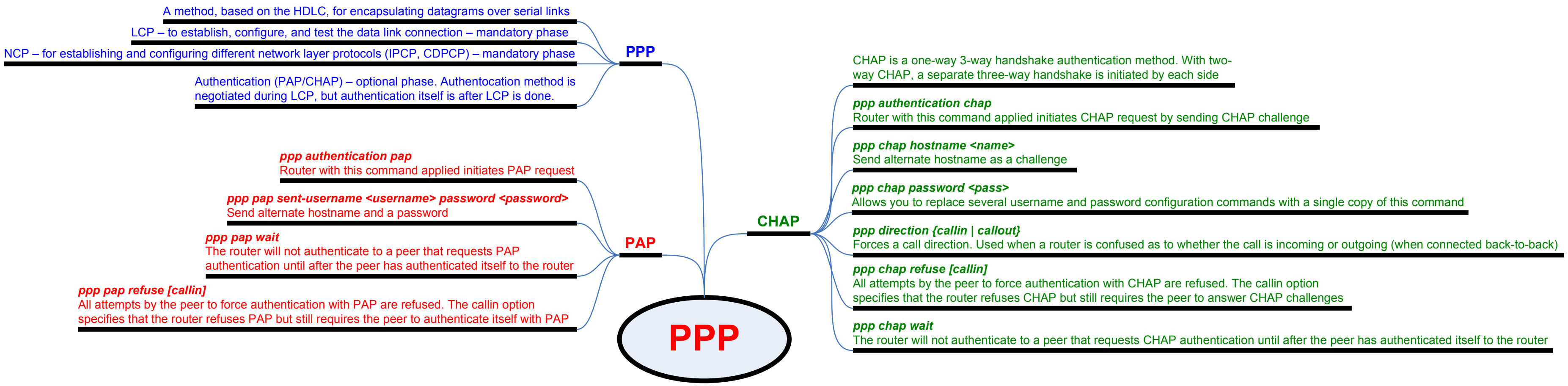
- **1)** The same DLCI on both sides
  - Disable LMI (***no keepalive***)
  - Router A and B:
    ***frame-relay interface-dlci 101***
- **2)** If DLCIs are to be different on both sides
  - Router A:
    ***frame-relay map ip <ip> 102*** (encapsulate)
    ***frame-relay interface-dlci 201*** (expect)
  - Router B:
    ***frame-relay map ip <ip> 201*** (encapsulate)
    ***frame-relay interface-dlci 102*** (expect)
- **3)** Frame-relay switching
  - ***keepalive*** must be enabled on both sides
  - Router A:
    ***frame-relay switching***
    ***frame-relay intf-type dce***
    ***frame-relay map ip <ip> 102***
    ***frame-relay interface-dlci 201***

## FR Autoinstall

- Router being configured will send BOOTP request for IP address over FR
- Staging router must have FR map configured
  ***fram-relay map ip <remote IP> <DLCI> broadcast*** (NBMA)
  ***frame-relay interface-dlci <dlci> protocol ip <ip>*** (P2P)
- 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

## PPPoFR

- Can be used to emulate p2p link on multipoint interface or to enable LFI on FRF.8 links (FR to ATM interworking)
- ***interface serial0/0***
  ***frame-relay interface-dlci <dlci> ppp virtual-template <id>***
  ***interface virtual-template <id>***
  ***ip address <ip> <mask> | ip unnumbered loopback0***
- Virtual-access interface is created after virtual-template is bound to DLCI. As 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)
- On multipoint interface each DLCI must be assigned to the same virtual-template interface because all endpoints must be in the same subnet. Separate virtual-access interface will be created for each DLCI
- ***interface multilink <ML-id>***
  ***ppp multilink***
  ***ppp multilink group <ML-id>***
  - ***interface virtual-template <VT-id>***
    ***ppp multilink group <ML-id>***



# PPP

## PPP

- A method, based on the HDLC, for encapsulating datagrams over serial links
- LCP – to establish, configure, and test the data link connection – mandatory phase
- NCP – for establishing and configuring different network layer protocols (IPCP, CDPCP) – mandatory phase
- Authentication (PAP/CHAP) – optional phase. Authentocation method is negotiated during LCP, but authentication itself is after LCP is done.

## PAP

- ***ppp authentication pap***
  Router with this command applied initiates PAP request
- ***ppp pap sent-username <username> password <password>***
  Send alternate hostname and a password
- ***ppp pap wait***
  The router will not authenticate to a peer that requests PAP authentication until after the peer has authenticated itself to the router
- ***ppp pap refuse [callin]***
  All attempts by the peer to force authentication with PAP are refused. The callin option specifies that the router refuses PAP but still requires the peer to authenticate itself with PAP

## CHAP

- CHAP is a one-way 3-way handshake authentication method. With two-way CHAP, a separate three-way handshake is initiated by each side
- ***ppp authentication chap***
  Router with this command applied initiates CHAP request by sending CHAP challenge
- ***ppp chap hostname <name>***
  Send alternate hostname as a challenge
- ***ppp chap password <pass>***
  Allows you to replace several username and password configuration commands with a single copy of this command
- ***ppp direction {callin | callout}***
  Forces a call direction. Used when a router is confused as to whether the call is incoming or outgoing (when connected back-to-back)
- ***ppp chap refuse [callin]***
  All attempts by the peer to force authentication with CHAP are refused. The callin option specifies that the router refuses CHAP but still requires the peer to answer CHAP challenges
- ***ppp chap wait***
  The router will not authenticate to a peer that requests CHAP authentication until after the peer has authenticated itself to the router

## PAP/CHAP Authentication

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

Client:

```
hostname R1

interface serial0/0
 ! Client sends username and password via PAP
 ppp pap sent-username R1 password cisco
```

Server:

```
hostname R2
username R1 password cisco

interface serial0/0
 ! server requests client to authenticate with PAP
 ppp authentication pap
```

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

Client:

```
hostname R1
username R2 password cisco

interface serial0/0
 ! Client sends username and password via PAP
 ppp pap sent-username R1 password cisco

 ! Client requests server to authenticate with CHAP
 ppp authentication chap
```

Server:

```
hostname R2
username R1 password cisco

interface serial0/0
 ! server requests client to authenticate with PAP
 ppp authentication pap

 ! server sends CHAP response using username R1
```

## Dynamic IP assignment

Client:

```
interface virtual-template 1
 ip address negotiated
```

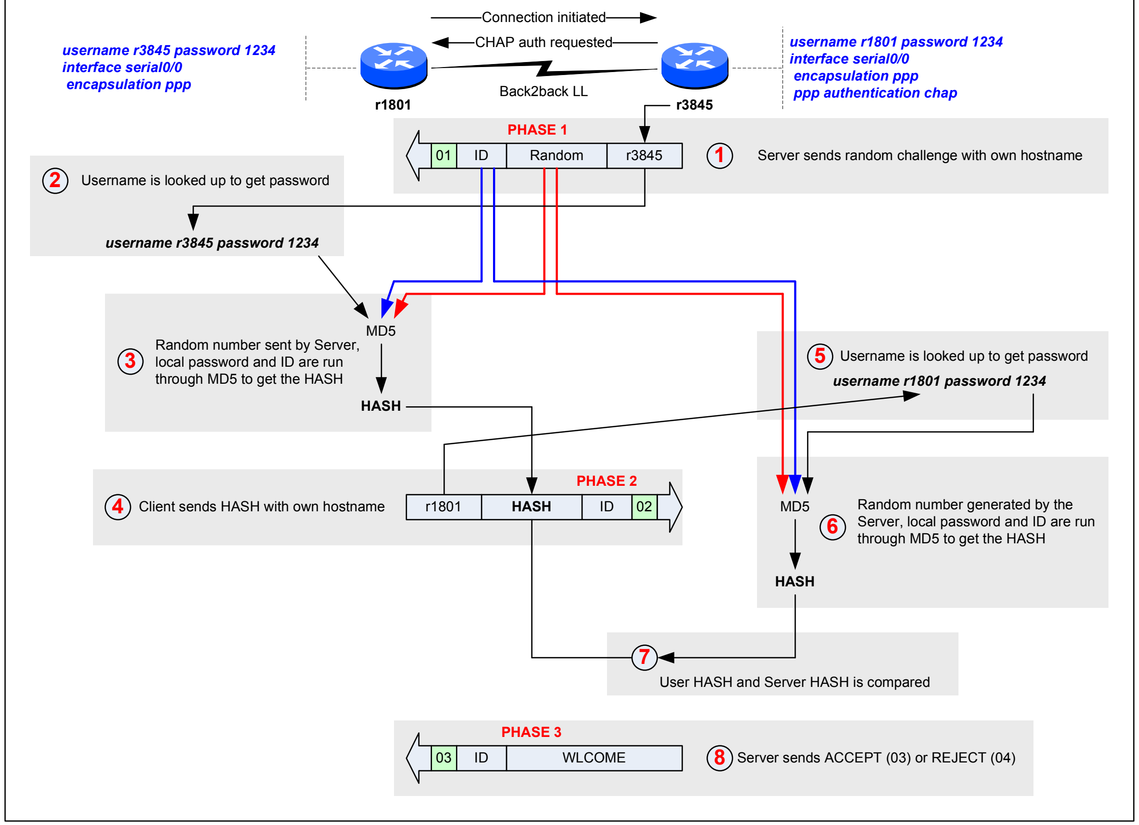
Server:

```
ip adress-pool local
ip local pool <name> <first IP> <last IP>

interface loopback 0
 ip address 10.0.0.1 255.255.255.255

interface virtual-template 1
 ip unnumbered loopback 0
 peer default ip address pool <name>
```

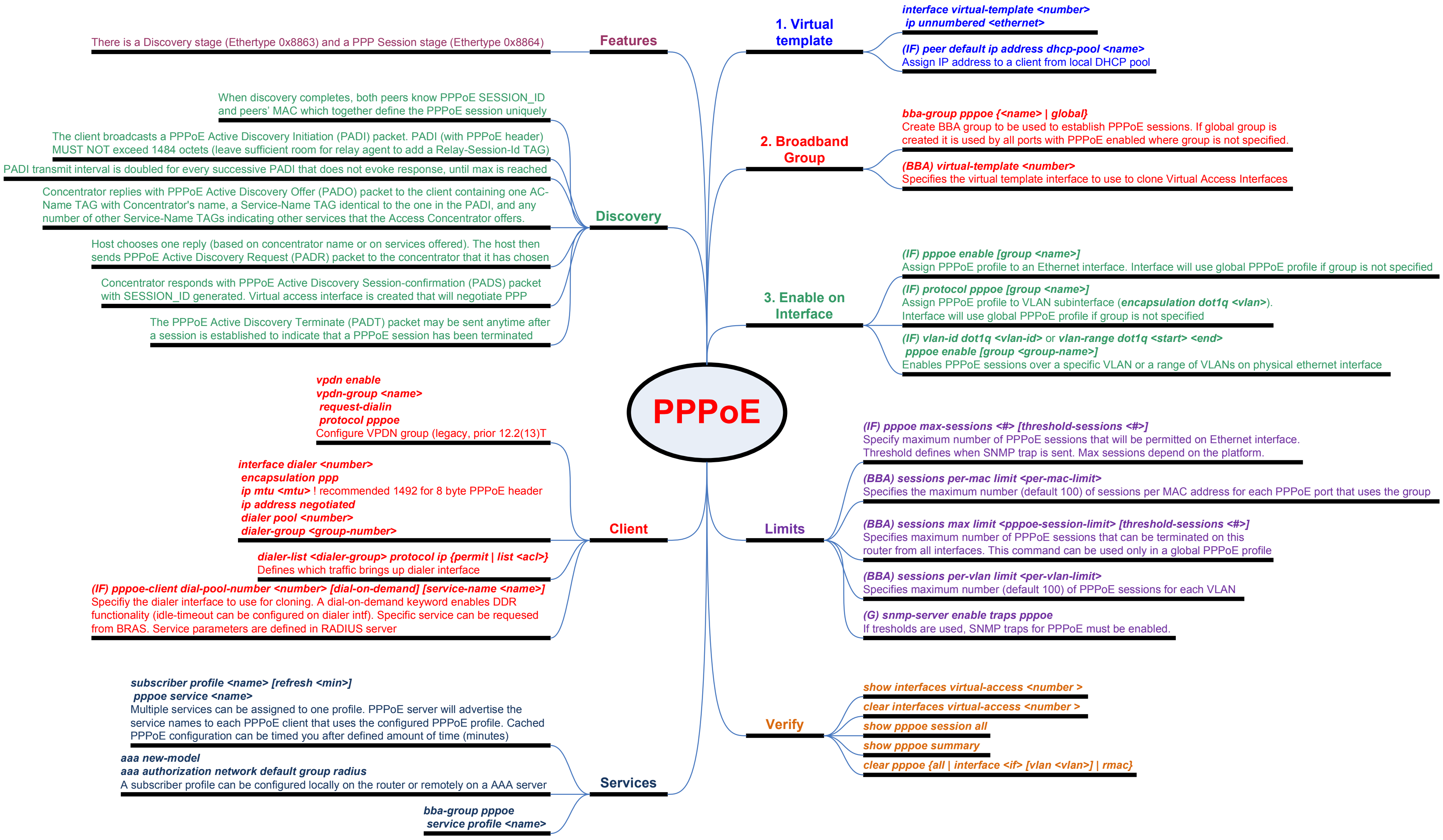
## CHAP Unidirectional 3-way challenge



```
username r3845 password 1234
interface serial0/0
 encapsulation ppp
```

```
username r1801 password 1234
interface serial0/0
 encapsulation ppp
 ppp authentication chap
```

1. Server sends random challenge with own hostname
2. Username is looked up to get password
   ***username r3845 password 1234***
3. Random number sent by Server, local password and ID are run through MD5 to get the HASH
4. Client sends HASH with own hostname
5. Username is looked up to get password
   ***username r1801 password 1234***
6. Random number generated by the Server, local password and ID are run through MD5 to get the HASH
7. User HASH and Server HASH is compared
8. Server sends ACCEPT (03) or REJECT (04)



# PPPoE

## Features

There is a Discovery stage (Ethertype 0x8863) and a PPP Session stage (Ethertype 0x8864)

## Discovery

When discovery completes, both peers know PPPoE SESSION_ID and peers' MAC which together define the PPPoE session uniquely

The client broadcasts a PPPoE Active Discovery Initiation (PADI) packet. PADI (with PPPoE header) MUST NOT exceed 1484 octets (leave sufficient room for relay agent to add a Relay-Session-Id TAG)

PADI transmit interval is doubled for every successive PADI that does not evoke response, until max is reached

Concentrator replies with PPPoE Active Discovery Offer (PADO) packet to the client containing one AC-Name TAG with Concentrator's name, a Service-Name TAG identical to the one in the PADI, and any number of other Service-Name TAGs indicating other services that the Access Concentrator offers.

Host chooses one reply (based on concentrator name or on services offered). The host then sends PPPoE Active Discovery Request (PADR) 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

## Client

***vpdn enable***
***vpdn-group <name>***
***request-dialin***
***protocol pppoe***
Configure VPDN group (legacy, prior 12.2(13)T

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

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

***(IF) pppoe-client dial-pool-number <number> [dial-on-demand] [service-name <name>]***
Specifiy 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

## 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 you after defined amount of time (minutes)

***aaa new-model***
***aaa authorization network default group radius***
A subscriber profile can be configured locally on the router or remotely on a AAA server

***bba-group pppoe***
***service profile <name>***

## 1. Virtual template

***interface virtual-template <number>***
***ip unnumbered <ethernet>***

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

## 2. Broadband Group

***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.

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

## 3. Enable on Interface

***(IF) pppoe enable [group <name>]***
Assign PPPoE profile to an Ethernet interface. Interface will use global PPPoE profile if group is not specified

***(IF) protocol pppoe [group <name>]***
Assign PPPoE profile to VLAN subinterface (***encapsulation dot1q <vlan>***).
Interface will use global PPPoE profile if group is not specified

***(IF) vlan-id dot1q <vlan-id>*** or ***vlan-range dot1q <start> <end>***
***pppoe enable [group <group-name>]***
Enables PPPoE sessions over a specific VLAN or a range of VLANs on physical ethernet interface

## Limits

***(IF) pppoe 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.

***(BBA) 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

***(BBA) 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

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

***(G) snmp-server enable traps pppoe***
If tresholds are used, SNMP traps for PPPoE must be enabled.

## Verify

***show interfaces virtual-access <number >***

***clear interfaces virtual-access <number >***

***show pppoe session all***

***show pppoe summary***

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

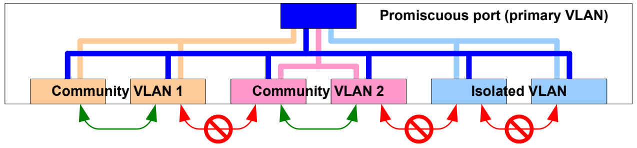
# VLAN

## VTP

- Mode
  - Transparent
    - VTP is disabled on the switch
  - Server
    - Does not propagate info untill domain is configured
  - Client
    - Can update server if revision is higher
- Advertises VLAN ID (1-1005), name, type, revision number only over Trunks
- By default, VTP operates in version 1. All switches must use the same version
- Initialy the switch is in VTP no-management-domain state until it receives an advertisement for a domain or domain is configured. If domain is learned next advertisements are ignored if revision number is lower
- If no domain is configured (Null) the first one heard is accepted, regardless of the mode (server and client). If domain is configured on the client it is also flooded among switches, so client can update server with domain name
- Every switch originates VTP summary every 5 min if no updates are heard and in response to VLAN change. Subset advertisement on vlan change (one per vlan)
- Enabling VTP pruning on a VTP server enables pruning for the entire management domain
- VTP pruning blocks unneeded flooded traffic to VLANs on trunk ports that are included in the pruning-eligible list. Vlans 2-1001 are pruning eligible
- ***(IF) switchport trunk prunning vlan <list>***
  List VLAN which are prune-eligible. Remaining VLANs will never be pruned
- ***(IF) switchport trunk allowed vlan <list>***
  Listed VLANs are not allowed to pass the trunk port, but are announced on that port. It can be used as a pruning mechanism on Transparent switches
- ***vtp interface loopback1 [only]***
  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)

## Private VLANs (3560)

- All hosts can be in the same subnet. **VTP transparent is required**
- When you enable DHCP snooping on primary VLAN, it is propagated to the secondary VLANs
- STP runs only on primary VLAN. Community and isolated VLANs do not have STP instance
- ***show vlan private-vlan***
- **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)
- Secondary
  - **community VLAN**
    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
  - **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



## QinQ Tuneling

- Tagged frames (Ethertype 0x8100) encapsulated within additional 4 byte 802.1q header (EtherType 0x88a8), so ***system mtu 1504*** must be added to all switches
- the native VLANs of the IEEE 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same switch
- Use the ***vlan dot1q tag native*** global command to configure the edge switch 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.
- Supports CDP, STP, MSTP, VTP, PAgP, LACP, and UDLD
- ***switchport mode dot1q-tunnel***
- ***l2protocol-tunnel [cdp | stp | vtp]***
- ***l2protocol-tunnel point-to-point [pagp | lacp | udld]***
  Tunnel etherchannel frames. Each pair of remote ports must be in different access VLAN
- ***l2protocol-tunnel cos <value>***

## Trunking

- DTP
  - Switches 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 choosen
  - Negotiation
    - ***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
    - ***switchport nonegotiate***
      Disable sending of DTP messages. Can be used only if trunking is configured.
- ISL
  - Cisco proprietary protocol supporting up to 1000 VLANs
  - SA is MAC of device doing trunking; DA is 0100.0c00.0000
  - Native (non-tagged) frames received from an ISL trunk port are dropped
  - Encapsulates in 26 bytes header and recalculated 4 bytes FCS trailer (real encapsulation) – total 30 bytes added to the frame
- 802.1q
  - IEEE standard for tagging frames on a trunk. Supports up to 4096 VLANs
  - Inserts 4 byte tag after SA and recalculates original FCS. Does not tag frames on the native VLAN

## Types

- Voice
  - If port is configured as access, the switch will automaticaly convert it internaly into a trunk
  - Portfast feature is automatically enabled when voice VLAN is configured
  - 802.1q frame
    - VLAN number is communicated to phone via CDPv2 (required for IPPhones)
    - ***switchport voice vlan <id>***
  - 802.1p frame
    - Switch treats frames with 802.1q tag set to zero as it was access port, but honors 802.1p COS field for prioritizing voice traffic. Traffic is then assigned to native VLAN
    - ***switchport voice vlan dot1p*** (VLAN 0)
- Native
  - Not supported on ISL trunks – all frames are tagged
  - ***vlan dot1q tag native***
    emulates ISL behaviour on 802.1q trunks for tagging native VLAN (required for QinQ)
  - On router subinterface – ***encapsulation dot1q <vlan-id> native***
  - On physical router interface – assumed if not configured on any subintf.
  - ***(IF) switchport trunk native vlan <id>***
  - When you remove VLAN 1 from a trunk port, the interface continues to send and receive management traffic (CDP, PAgP, LACP, DTP, VTP) within VLAN 1.
- Normal range 1-1005
  - Can be configured in Server and Transparent modes
- Extended range 1006 - 4096
  - The VLAN database configuration mode (***vlan database***) 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 (***show vlan internal usage***)

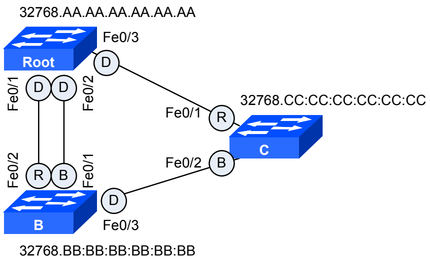
## VMPS

- ***(IF) switchport access vlan dynamic***
- Client talks to server with VLAN Query Protocol (VQP)
- When configured as secure mode the port is shutdown if MAC-to-VLAN mapping is not in database. Otherwise, access is denied but port stays up
- 3560 can be a client and a server. 3550 can be a client only
- ***vmps reconfirm <sec>*** - default refresh is every 60 min
- ***vmps retry <#>*** - default 3 times
- ***vmps server <ip> [primary]***

# Cisco PVST+

## Timers Features

- PVST was supported only on ISL trunks
- Bridges are not interested in local timers, they use timers send by Root Hellos.
- Blocking => Listening (15sec) => Learning (15 sec) => Forwarding
- ***spanning-tree vlan <id> hello-time <sec>*** (default is 2 sec)
- ***spanning-tree vlan <id> forward-time <sec>*** (default is 15 sec)
- ***spanning-tree vlan <id> max-age <sec>*** (default is 20 sec)
  Bridge waits 10 Hello misses before performing STP recalculation
- Each bridge adds 1 hop (second) to BPDU age, 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



| Byte 2 | | | | | | | | Byte 1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Priority | | | | Extended System ID (VLAN ID) | | | | | | | | | | | |
| 32768 | 16384 | 8192 | 4096 | 2048 | 1024 | 512 | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

That's why priority is in multiples of 4096

## 1. Elect the Root bridge

- **Lowest Priority (Priority+VLAN+MAC) wins root election**
  - Priority – 2 bytes 32768 (0x8000)
    - 4 bits configurable Priority (multiple of 4096)
    - 12 bits System ID Extension – VLAN ID. Allows different Roots per VLAN (802.1t STP extension)
  - ID – 6 bytes MAC
- If superior (lowest) Hello is heard, own is ceased. Superior is forwarded
- ***(G) spanning-tree vlan <id> priority <0-61440>***
- ***(G) spanning-tree vlan <id> root {primary|secondary} [diameter <hop#>]***
  - ***primary***: 24576 or 4096 less than existing one (macro listens to root BPDUs)
  - ***secondary:*** 28672
  - ***diameter:*** causes changes to Hello, Forward delay and Maxage timers
- Each switch forwards root's Hello changing some fields
  - Cost (total cost to the Root) – added from interface on which BPDU was received. Can be manipulated with BW, speed, and manualy set per VLAN on intf.
  - Forwarder's ID
  - Forwarder's port priority – configured on interface out which BPDU is sent
  - Forwarder's port number – outgoing interface

## 2. Determine Root Port

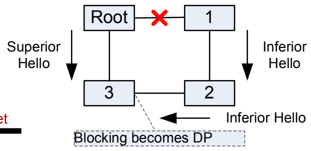
- **1.** Port on which Hello was received with lowest Cost (after adding own cost)
  - 10Mb – 100
  - 100Mb – 19
  - 1Gb – 4
  - 10Gb – 2
- **2.** Lowest forwarder's Bridge ID – the one who sent BPDU to us
- ***(IF) spanning-tree vlan <id> cost <path-cost>*** (configured on root port)
- **3.** Lowest forwarder's (peer's) port priority (default is 128, 0 to 240 in increments of 16)
- ***(IF) spanning-tree vlan <id> port-priority <0-250>*** (configured on designated port)
- **4.** Lowest forwarder's port number

## 3. Determine Designated Ports

- Only one switch can forward traffic to the same segment
- Hellos with lowest advertised cost (without adding own cost) becomes DP
- Switch with inferior Hellos stops forwarding them to the segment
- If advertised costs are the same the tiebreaker is exactly the same as for RP
  - 1. Lowest peer's Bridge ID
  - 2. Lowest peer's port priority
  - 3. Lowest peer's port number

## 4. Topology change

- If 10 Hellos are missed (Maxage 20 sec) each switch thinks it is a root and starts sending own Hellos again
- If another switch receives this Hello on blocking port, and it hears superior Hello on different port, it switches over from blocking to DP and starts forwarding superior Hellos
- All switches need to be informed about the change to timeout CAM
  - Switch sends TCN BPDU 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 TC in Hellos for that time.



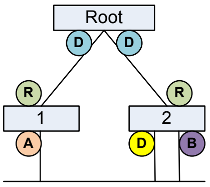
# Rapid 802.1w

## Features

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

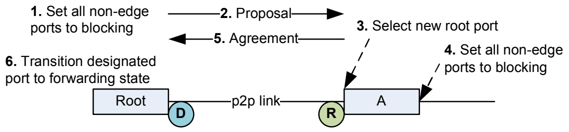
## Port roles

- New port roles used for fast convergence
  - **B**ackup port – on the same switch
  - **A**lternate port – on different switch
- Port types
  - **point-to-point**
    - Between switches (FDX port)
    - ***spanning-tree link-type point-to-point***
      The p2p state can be manualy forced if HDX (half-duplex) is used
  - **Shared**
    - Where HUB is connected (HDX)
  - **Edge**
    - ***spanning-tree portfast***



## Convergence

- Topology change
  - If topology change is detected, switch sets a TC timer to twice the hello time and sets the TC bit on all BPDUs sent out to its designated and root ports until the timer expires
  - If switch receives a TC BPDU, it clears the MAC addresses on that port and sets the TC bit on all BPDUs sent out its designated and root ports until the TC timer expires
- Sync
  - Upstream bridge sends a proposal out of DP (sets proposal bit in outgoing BPDU)
  - Downstream bridge blocks all non-designated ports and authorizes upstream brodge to put his port into forwarding state



1. Set all non-edge ports to blocking
2. Proposal
5. Agreement
3. Select new root port
4. Set all non-edge ports to blocking
6. Transition designated port to forwarding state
Root
D
p2p link
R
A

# MST 802.1s

## Features

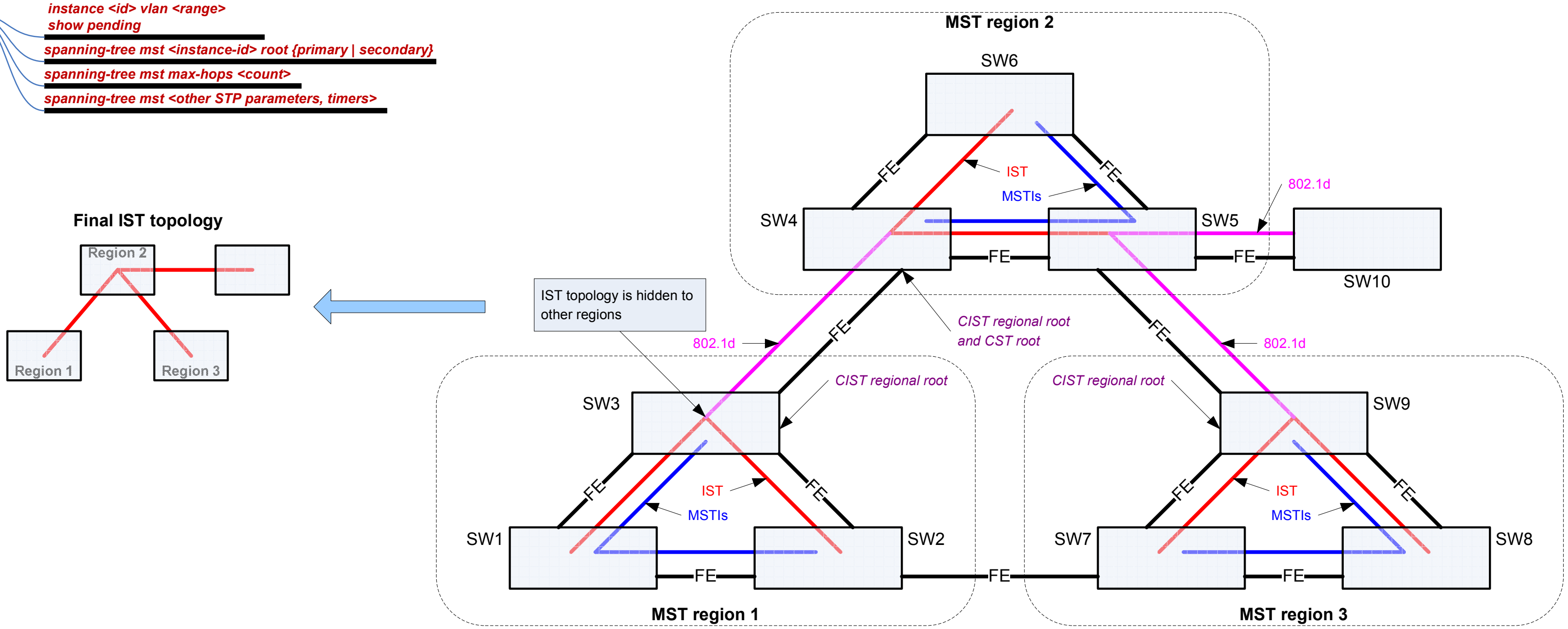
- Up to 16 MST instances (no limit for VLANs) – there is always one instance: 0
- All switches within a region must have identical configuration (different configuration means different region)
- VLAN-to-instance mapping is not propagated with BPDU. Only digest with region name and revision number is 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 hops 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

## Instances

- **IST (MSTI 0)**
  Internal Spanning Tree
  - The only instance that sends and receives BPDUs. All of the other STP instance information is contained in M-records, which are encapsulated within MSTP BPDUs
  - MST Region replicates IST BPDUs within each VLAN to simulate PVST+ neighbor
  - Represents MST region as CST virtual bridge to outside
  - RSTP instance that extends CST inside region
  - By default, all VLANs are assigned to the IST.
  - STP parameters related to BPDU transmission (hello time, etc) are configured only on the CST instance but affect all MST instances. However, each MSTI can have own topology (root bridge, port costs)
- **MSTI** – Multiple Spanning Tree Instances (one or more) - RSTP instances within a region. RSTP is enabled automatically by default
- **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
  - Each region selects own CIST regional root. It must be a boundary switch with lowest CIST external path cost
  - External BPDUs are tunneled (CIST metrics are passed unchanged) across the region and processed only by boundary switches.
  - When switch detects BPDU 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

## Configuration

- ***(G) spanning-tree mode mst***
- ***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>***



# PortChannel

## Cisco PAgP

- Up to eight compatibly configured Ethernet interfaces
- *(IF) channel-protocol pagp*
- *(IF) channel-group <1-64> mode {auto | desirable} [non-silent]*
  In silent mode etherchannel can be built even if PAgP packets are not received.
  The silent setting is for connections to file servers or packet analyzers
- *(G) pagp learn-method {aggregation-port | physical-port}*
- *(IF) 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

| Cisco | 802.1d | Behaviour |
|---|---|---|
| on | on | No dynamic negotiation. Forced. |
| off | off | PortChannel disabled |
| auto | passive | Wait for other side to initiate |
| desirable | active | Initiate negotiation |

## IEEE 802.3ad LACP

- 16 ports can be selected, but only max 8 is used. Rest is in standby (LACP port-priority and Port ID decide which are standby; lower is better)
- Switch with lowest system priority makes decisions about which ports participate in bundling (switch used port-priorities)
- *(IF) channel-protocol lacp*
- *(IF) channel-group <1-64> mode {passive | active}*
- *(IF) lacp port-priority <#>* (default 32768, lower better)
- *(G) lacp system-priority <#>* (lower better)
- *show lacp sys-id*

## Load balancing

- *(G) port-channel load-balance {src-mac | dst-mac}*
  XOR on rightmost bits of MAC

# Port Protection

## BPDU guard

- err-disable portfast port upon receiving BPDU
- *(G) spanning-tree portfast bpduguard default*
- *(IF) spanning-tree bpduguard enable*

## BPDU filter

- Supported on PVST+, rapid-PVSTP+ or MST
- *(G) spanning-tree portfast bpdufilter default*
  portfast port switches to non-portfast upon receiving BPDU
- *(IF) spanning-tree bpdufilter enable*
  it does not send any BPDUs and drops all BPDUs it receives

## 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 err-disable
- *(G) spanning-tree etherchannel guard misconfig*

## Root guard

- Can be enabled on **designated ports only**. Opposite to loop guard
- Ignores superior Hellos received on a user port (root-inconsistent)
- Cannot be configured on backup ports when uplinkfast is configured
- Applies to all the VLANs to which the interface belongs
- *(IF) spanning-tree guard root*

## Loop guard

- If no BPDUs are received on a blocked port for a specific length of time Loop Guard puts that port (per VLAN) into loop-inconsistent blocking state, rather than transitioning to forwarding state
- Can be enabled on **non-designated ports only**
- Automatic recovery if BPDU is received
- *(G) spanning-tree loopguard default*
- *(IF) spanning-tree guard loop*

## UDLD

- fiber and copper (copper uses Link Pulses, so not so susceptible)
- Uses L2 probes every 15 sec to mac 01:00:0C:CC:CC:CC. Must be ACKed by remote end.
- *(G) udld message time <sec>* - frequency of probes
- Sends local port ID and remote (seen) port ID. Remote end compares with own state
- Normal mode does nothing except syslog
- Aggresive mode attempts to reconnect once a second 8 times before err-disabling
- If configured for the first time it is not enabled untill first Hello is heard
- *(IF) udld enable*
- *(IF) udld port aggressive* – For fiber and UTP links
- *(G) udld {aggressive | enable}*
  Affects fiber connections only

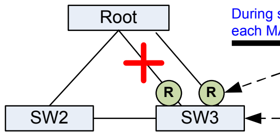
# Convergence

## 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*
- *(IF) spanning-tree portfast*

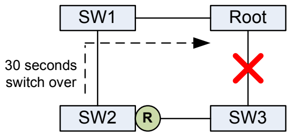
## Uplinkfast

- Used on access switch with multiple uplinks to core
- Priority is automaticaly set to 49152 so the switch will not become root. Port cost is set to 3000 so it will not transit any traffic
- During switchover to new RP, for each connected MAC it multicasts frame with each MAC as SA forcing other switches to update CAM. Other MACs are cleared
- 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)



## Backbonefast

- *(G) spanning-tree backbonefast*
- Indirect link failure detection. recovery within 30 sec.
- All switches within a domain must be configured
- If first Hello is missed switch sends Root Link Quety (RLQ) out the port where Hello was expected. If neighbor switch lost previous root too (roots are compared for the switch and the neighbor), it informes that switch and re-convergence (STP) occurs without waiting for Maxage timeout (20 sec)



# SPAN

## SPAN

- 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.
- Receive (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 od QOS drop.
- Transmit (Tx) SPAN – catch frames after all modification and processing is performed by the switch. In the case of output ACLs, if the SPAN source drops the packet, the SPAN destination would also drop the packet
- ***monitor session 1 source vlan 5 rx***
- ***monitor session <#> filter vlan <vlan-ids>*** (Limit the SPAN source traffic to specified VLANs)
- ***monitor session 1 source interface fastethernet0/1 [rx | tx | both]***
- ***monitor session 1 destination interface fastethernet0/8***

## RSPAN

- You cannot use RSPAN to monitor Layer 2 protocols (CDP, VTP, STP)
- 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
- ***vlan <id>***<br>***remote-span*** (on source switch only, remote switch will learn this information)
- ***SW1: monitor session 1 source interface fastethernet0/1 [rx | tx | both]***
- ***SW1: monitor session 1 source vlan 5 rx***
- ***SW1: monitor session 1 destination remote vlan 901 reflector-port fastethernet0/1***
- ***SW2: monitor session 1 source remote vlan 901***
- ***SW2: monitor session 1 destination interface fastethernet0/5***

**Common Protocol Types**

| | |
|---|---|
| 802.1q | 0x8100 |
| ARP | 0x0806 |
| RARP | 0x8035 |
| IP | 0x0800 |
| IPv6 | 0x86DD |
| PPPoE | 0x8863/0x8864 |
| MPLS | 0x8847/0x8848 |
| IS-IS | 0x8000 |
| LACP | 0x8809 |
| 802.1x | 0x888E |

**Ethernet starndards**

| | |
|---|---|
| IEEE 802.2 | LLC |
| IEEE 802.3u | FE 100Mbps |
| IEEE 802.3z | GE 1000Mbps Optical |
| IEEE 802.3ab | GE 1000Mbps Copper |
| IEEE 802.3ae | 10GE |

# Macro

## Smartport

- ***macro name USER_PORT***<br>***switchport mode access***<br>***switchport access vlan $vlanID***<br>***spanning-tree portfast***
- ***(IF) macro apply USER_PORT $vlanID 10***
- After applying macro to interface or to global config, ***macro description <name>*** will be added

## Range

- ***interface range macro <name>***
- ***define interface-range <name> <intf range>***

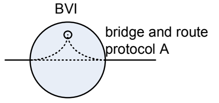
# Bridging

## Transparent

- Complies with the IEEE 802.1D standard
- ***bridge <bridge-group> protocol ieee***
- ***(IF) bridge-group <bridge-group>***
- ***bridge <bridge-group> address <mac-address> {forward | discard} [<intf>]***

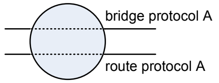
## IRB

- Integrated routing and bridging makes it possible to route a specific protocol between routed interfaces and bridge groups, or route a specific protocol between bridge groups
- The bridge-group virtual interface is a normal routed interface that does not support bridging, but does represent its corresponding bridge group to the routed interface
- Packets coming from a routed interface, but destined for a host in a bridged domain, are routed to BVI and forwarded to the corresponding bridged interface
- All routable traffic received on a bridged interface is routed to other routed interfaces as if it is coming directly from BVI.
- ***bridge irb***
- ***interface bvi <bridge-group>***
- ***bridge <bridge-group> route <protocol>***
- ***bridge <bridge-group> bridge <protocol>***



## CRB

- Route a given protocol among one group of interfaces and concurrently bridge that protocol among a separate group of interfaces
- Protocol may be either routed or bridged on a given interface, but not both
- When CRB is enabled, you must configure explicit bridge route command for any protocol that is to be routed on the interfaces in a bridge group
- ***bridge <bridge-group> route <protocol>***
- ***bridge crb***



# 35x0 Features

## FlexLink

- 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
- A backup link does not have to be the same type
- STP is automaticaly 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
- ***(IF) switchport backup interface <intf>***
- ***(IF) switchport backup interface <intf> preemption mode [forced | bandwidth | off]***<br>forced – active always preempts; bandwidth - intf with higher BW always acts as active
- ***(IF) switchport backup interface <intf> preemption delay <sec>*** (default 35 sec)
- ***(IF) switchport backup interface <intf> mmu primary vlan <vlan-id>***<br>If not defined, the lowest VLAN is used for MAC-address move updates
- ***(G) mac address-table move update transmit***<br>Enable the access switch to send MAC address-table move updates to other switches
- ***(G) mac address-table move update receive***<br>Enable the switch to get and process the MAC address-table move updates

## MAC notification

- ***(G) snmp-server enable traps mac-notification***
- ***(G) mac address-table notification change***
- ***mac address-table notification change [history-size <#>] [interval <sec>]***<br>By default traps are sent every 1 sec. History size is 1.
- ***(IF) snmp trap mac-notification {added | removed}***

## Fallback bridging

- With fallback bridging, the switch bridges together two or more VLANs or routed ports, essentially connecting multiple VLANs within one bridge domain
- Fallback bridging does not allow spanning trees from VLANs to collapse. Each VLAN has own SPT instance and a separate SPT, called VLAN-bridge SPT, which runs on top of the bridge group to prevent loops
- ***bridge <bridge-group> protocol vlan-bridge***
- ***(IF) bridge-group <bridge-group>***
- By default, switch forwards any frames it has dynamically learned. But, the switch only forward frames whose MAC addresses are statically configured (static MAC for bridge, not for mac-address-table !!!).
- ***1) no bridge <group> acquire***<br>***2) bridge <group> address <mac> {forward | discard} [<interface>]***

# NTP

## Access control

- Control messages – reading and writing internal NTP variables
- Request/Update messages – actual time synchronization
- ***ntp access-group {query-only | serve-only | serve | peer} <acl>***
  If multiple ACLs are used, requests are scanned in the following order:
  ***peer*** – accept and reply to clock updates and control messages
  ***serve*** – only reply to clock requests and control messages
  ***serve-only*** – reply only to clock requests
  ***query-only*** – reply only to control messages
- ***(IF) ntp disable***
  Stop sending and responding to NTP messages on that interface
- ***ntp source <intf>***
  Source of NTP messages

## Server

- ***ntp master [<stratum>]***
  If stratum is omited, 8 is used. Each peer using server adds 1 to stratum
- Internal server is created, running on 127.127.7.1. This IP must be explicitly allowed by ***ntp access-group peer <acl>***, if ACLs are used.

## Client

- Client is only going to synchronize its clock to another NTP clock source
- ***ntp server <ip> [<ver>] [key <key>] [source <if>] [prefer]***
- A client can act as a server, serving another clients (cascading queries)
- Queries are sent every 60 seconds.

## Symetric active mode

- Create a peer association if this router is willing to synchronize to another device or allow another device to synchronize to itself
- ***ntp peer <ip> [<ver>] [key <key>] [source <if>] [prefer]***

## Broadcast

- **Server:**
  ***(IF) ntp broadcast***
- **Client:**
  ***(IF) ntp broadcast client***

## Authentication

- **Client authenticates the server ONLY !!!**
- **Client:**
  ***ntp server <ip> [key <key>]***
  ***ntp authenticate***
  ***ntp authentication-key <id> md5 <password>***
  ***ntp trusted-key <id>***
- **Server:**
  ***ntp authentication-key <id> md5 <password>***
  only this is required to send the key to client. Key ID and password must match the one requested by the client (client sends key ID with a request)

# CDP

- ***cdp timer <sec>*** - CDP messages advertisement interval (default 60 sec)
- ***cdp holdtime <sec>***
  inform receiving device, how long CDP messages should be stored localy (default 180)
- ***(G) cdp run***
  ***(IF) cdp enable***
- ***no cdp log mismatch duplex***
  Duplex mismatches are displayed for all Ethernet interfaces by default.
- ***cdp source-interface <if>***
  IP from this interface will be used to identify device (messages will be originated from this intf). It should not be an IP unnumbered interface.
- CDP runs on any media that supports the subnetwork access protocol (SNAP)

# ARP

## Features

- Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet use Subnetwork Access Protocol (SNAP).
- ***(IF) arp timeout <sec>*** - default is 4 hours
- ***arp <ip-address> <hardware-address> arpa [<interface>]***

## RARP

- RARP only provides IP addresses of the hosts and not subnet masks or default gateways
- Reverse ARP (RARP) requests an IP address instead of a MAC address. RARP often is used by diskless workstations because this type of device has no way to store IP addresses to use when they boot.

## Proxy ARP

- Proxy ARP is enabled by default
- ***(IF) no ip proxy-arp***
- ***(G) ip arp proxy disable***
- ***ip local-proxy-arp***
  Port replies to ARP requests on the local segment to allow communication between protected ports.

- ***(IF) ip gratuitous-arps***
  Gratuitous ARP - A host might occasionally issue an ARP Request with its own IPv4 address as the target address to check duplicate addresses. disabled by default

## Local Area Mobility (LAM)

- ***(IF) ip mobile arp access-group <acl>***
  Router starts to listen to ARPs from hosts which are not in the same subnet as on interface. Then host's IP is installed in routing table as /32. ACL defines for which IPs to listen to.
- ***router <protocol>***
  ***redistribute mobile metric 1***

## Secure ARP

- ***(IF) arp authorised***
  disable dynamic Address Resolution Protocol (ARP) learning on an interface. Mapping of IP address to MAC address for an interface can be installed only by the authorized subsystem or static entries
- ***arp probe internal <sec> count <#>***
  Probing of authorized peers.
- ***ip dhcp pool <name>***
  ***update arp***
  Used to secure ARP table entries and their corresponding DHCP leases (only new ones, existing remain unsecured untill lease time expires)
- The ***clear arp-cache*** will not remove secure arp entries, ***clear ip dhcp binding*** must be used

# WCCP

## Features

- WCCP works only with IPv4 networks. Uses UDP/2048
- Up to 32 Content Engines for a router in WCCPv1. CE with lowest IP is elected as leading Content Engine
- WCCPv1 supports only HTTP (port 80) traffic
- In WCCPv2 (default) there can be more than one router serving Content Engine cluster
- WCCPv2 supports MD5 authentication and load distribution
- When WCCP forwards traffic via GRE, the redirected packets are encapsulated within a GRE header, and a WCCP redirect header. When WCCP forwards traffic using L2 (Cache Engine is on the same segment as the router), the original MAC header of the IP packet is overwritten and replaced with the MAC header for the WCCP client.

## Configuration

- ***ip wccp web-cache*** (enable WCCP)
- ***ip wccp web-cache group-address <multicast> password <pass>***
- ***ip wccp web-cache redirect-list <acl>*** - for which clients redirection is enabled
- ***ip wccp web-cache group-list <acl>*** - which cache engines are allowed to participate
- ***(IF) ip wccp web-cache redirect out*** (select interface toward Internet)
- ***(IF) ip wccp redirect exclude in*** – exclude interface from redirecion
- ***ip wccp mode {open | closed}***
  When closed mode is enabled, and a content engine is not available, all traffic which would normaly be passed through it, is blocked

# Routing features

## Network classes

| | |
|---|---|
| 1 – 126 | A |
| 127 | Loopback |
| 128 – 191 | B |
| 192 – 223 | C |
| 224 – 239 | D |
| 240 – 255 | Reserved |

## Protocol #

| | |
|---|---|
| 1 | ICMP |
| 2 | IGMP |
| 4 | IP |
| 6 | TCP |
| 17 | UDP |
| 41 | IPv6 |
| 46 | RSVP |
| 47 | GRE |
| 50 | ESP |
| 51 | AH |
| 88 | EIGRP |
| 89 | OSPF |
| 103 | PIM |
| 112 | VRRP |

## Administrative Distance

| | |
|---|---|
| Directly connected | 0 |
| Static to interface | 0 |
| Static to NH | 1 |
| EIGRP Summary | 5 |
| eBGP | 20 |
| EIGRP Internal | 90 |
| IGRP | 100 |
| OSPF | 110 |
| ISIS | 115 |
| RIP | 120 |
| EGP | 140 |
| ODR | 160 |
| EIGRP external | 170 |
| iBGP | 200 |
| BGP local | 200 |
| Unknown (not valid) | 255 |

## Port numbers

| | |
|---|---|
| echo | 7/tcp/udp |
| discard | 9/tcp/udp |
| daytime | 13/tcp/udp |
| chargen | 19/tcp/udp |
| bootps | 67/tcp/udp |
| bootpc | 68/tcp/udp |
| auth | 113/tcp/udp |
| ntp | 123/tcp/udp |
| netbios-ns | 137/tcp/udp |
| netbios-dgm | 138/tcp/udp |
| netbios-ssn | 139/tcp/udp |
| snmp | 161/tcp/udp |
| snmptrap | 162/tcp/udp |
| bgp | 179/tcp |
| syslog | 514/udp |
| shell | 514/tcp |
| rip | 520/udp |
| ripng | 521/tcp/udp |

## Redistribution

**Step 1:** get all routes which are in routing table and belong to redistributed protocol (***show ip route <protocol>***)

**Step 2:** get all connected routes which are covered by redistributed protocol with network command (***show ip route connected <addr>*** *-> redistributed by <protocol>*)

Chain distribution on one router is **<u>NOT</u>** possible. Ex. EIGRP -> RIP -> OSPF, EIGRP routes will be redistributed into RIP, but NOT into OSPF.

Routes redistributed from one protocol (higher AD) into another protocol (lower AD) will NOT be in the routing table on redistributing router as originated by the second protocol, although AD is lower. Route to be redistributed must be in the routing table, so it could cause endless reditribution loop

## Advanced Object Tracking

### Reliable routing (Conditional default route injection)

**1**. Track remote router with RTR:
***track 1 rtr 1 reachability***
***delay down <sec> up <sec>***

**2**. Create bogus static routing, reacting 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 recource is UP:
***ip route 1.1.1.1 255.255.255.255 null 0 track 1***

**3**. Create prefix-list covering bogus route and assign it to route-map
***ip prefix-list TST permit 1.1.1.1/32***
***route-map TST permit 10***
***match ip address prefix-list TST***

**4.** Originate a default route (RIP in this example) only if route-map result is true, meaning the remote router is reachable:
***router rip***
***default-information originate route-map TST***

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 1 <u>not</u>***
***object 2***

***track timer interface <sec>*** (default is 1 sec)
***track timer ip-route <sec>*** (default is 15 seconds)

## ODR

hub router can automatically discover stub networks while the stub routers still use a default route to the hub (also learned via ODR: ***0* 0.0.0.0 [160/1] via ...***)

ODR conveys only the network portion of the address

It discovers information about stub networks but does not provide any routing information to the stub routers. Information is conveyed by a CDP

The metric (hop count) will never be more than 1

CDP runs on any media that supports the subnetwork access protocol (SNAP), which means that ODR also depends on SNAP support.

Hello 60sec, Invalid 180sec. ODR advertisements stop if any other protocol runs on stub

Hub: ***router odr***

## Distribute-list

When using extended ACL in distribute-list in IGP, the „source" part is an update source of the route, and „destination" is network to be matched (distributed)
***router <IGP-protocol>***
***distribute-list <ext acl> {in | out} <intf>***
***access-list <ext acl> permit ip <source> <mask> <network> <mask>***

***ip access-list resequence <acl> <start> <step>***
Resequence ACL. By default each entry is seqenced by 10, starting with 10

***distribute-list prefix <prefix1 name> gateway <prefix2 name> {in | out}***
Filter prefixes in prefix1 list received from gateways listed in prefix2 list

## Distance

If AD is manipulated, and two protocols have the same AD, the tie-breaker is the default, original AD for each protocol

***distance <distance> <ip> <mask> <acl>***
ip/mask – advertising router
acl – which routes will get new distance

## PBR

***(IF) ip policy route-map <name>***
Affects incoming packets only

***(IF) ip route-cache same-interface***
May be required if next-hop points to the same interface (ex. NBMA)

***set ip next-hop <ip> verify-availability***
Verify the availability of the next-hop address before attempting to forward the packet. The router will search CDP table to verify that the next-hop address is listed

***set ip next-hop <ip> track <id>***
next hop can be also tracked with Advanced Object Tracking. There can be many next hops defined in one route-map entry. If one fails, the next one is checked.

***ip local policy route-map <name>***
for traffic originated by the router. It can be usefull to pass router-generated traffic through ACL or CBAC. By default router-generated traffic does not pass any outbound ACLs.

## Route-map

If a route is denied by ACL in „permit" statement it doesn't mean route is not redistributed at all, it's just not matched by this entry

There is IMPLICIT DENY at the end of route-map

if no action or sequence number is specified when the route map is configured, the route map will <u>default to a permit</u> and a sequence number of 10

### Continue

Jump to specified seq or next seq if seq is not specified

If match clause exists, continue proceeds only if match is successful

***continue <seq>***

If next RM entry (pointed by continue) also have continue clause but match does not occur, second continue is not processed, and next RM entry is evaluated

## Backup interface

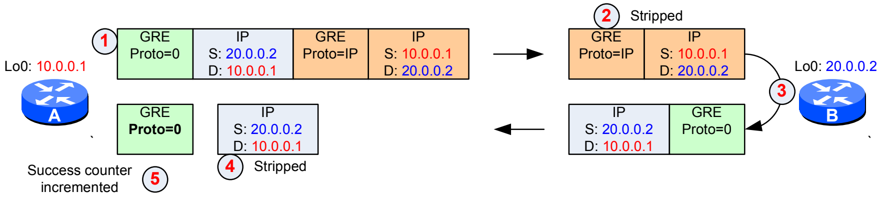
***(IF) backup interface <backup-intf>***
The interface defined with this command can back up only one other interface. The backing up interface goes into standby mode and cannot be used to carry any traffic until activated.

***backup delay {<enable-delay> | never} {<disable-delay> | never}***
To immediately switchover to backup interface specify delay = 0

## GRE

Protocol number 47

***(IF) keepalive <sec> <retry count>***
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



Lo0: 10.0.0.1
A
1
GRE Proto=0
IP S: 20.0.0.2 D: 10.0.0.1
GRE Proto=IP
IP S: 10.0.0.1 D: 20.0.0.2
2
Stripped
GRE Proto=IP
IP S: 10.0.0.1 D: 20.0.0.2
Lo0: 20.0.0.2
B
3
IP S: 20.0.0.2 D: 10.0.0.1
GRE Proto=0
GRE Proto=0
IP S: 20.0.0.2 D: 10.0.0.1
4
Stripped
Success counter incremented
5

## Match Classes

Class A: ***ip prefix-list A permit 0.0.0.0/1 ge 8 le 32 <=> access-list 100 permit 0.0.0.0 127.255.255.255***

Class B: ***ip prefix-list B permit 128.0.0.0/2 ge 16 le 32 <=> access-list 100 permit 128.0.0.0 63.255.255.255***

Class C: ***ip prefix-list C permit 192.0.0.0/3 ge 24 le 32 <=> access-list 100 permit 192.0.0.0 31.255.255.255***

# OER/PfR Basics

## Features

- Communication between MC and BR – UDP/3949, TCP/3949
- Traditional routing uses static metrics and destination-based prefix reachability. Network recovery is based on neighbor and link failures. PfR enchances routing to select the best path based on measurements and policy
- OER monitors traffic class performance and selects the best entrance or exit for traffic class. Adaptive routing adjustments are based on RTT, jitter, packet loss, MOS, path availability, traffic load and cost policy
- Minimum CPU impact. Utilizes lot's of memory (based on prefixes). MC is the most impacted.
- The preferred route can be an injected BGP route or an injected static route
- PfR is a successor of OER. OER provided route control on per destination prefix basis. PfR expandeds capabilities that facilitate intelligent route control on a per application basis
- OER can learn both outside and inside prefixes.

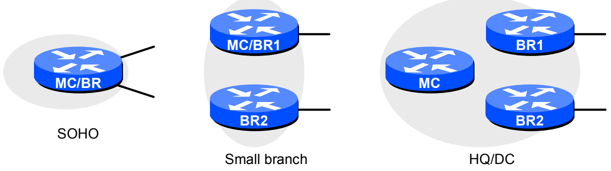
## Master Controller

### Features

- Verifies that monitored prefix has a parent route with valid next hop before it asks BR to alter routing
- Does not have to be in forwarding path, but must be reachable by BRs
- Long-term stats are collected every 60 min. Short-term stats are collected every 5 min
- Monitors the network and maintains a central policy database with statistics
- Support up to 10 border routers and up to 20 OER-managed external interfaces
- MC will not become active if there are no BRs or only one exit point exists
- Can be shutdown with ***shutdown*** command

### Config

- ***oer master***
  Enable OER master controller
- ***border <ip> [key-chain <name>]***
  At least one BR must be configured. Key chain is required when adding BR for the first time. It's optional when reconfiguring existing BR
- ***interface <if> {external | internal}***
  Define interfaces which are used on BR (must exist on BR)
- ***port <port>***
- ***logging***
  Enables syslog messages for a master controller (*notice* level)
- ***keepalive <sec>***
  Keepalive between MC and BR. Default is 60 sec.



## Border Router

### Features

- Edge router with one or more exit links to an ISP or WAN
- Enforces policy changes so it must be in the forwarding path
- Reports prefix and exit link measurements to MC
- Can be enabled on the same router as a MC
- ***interface virtual-template 1***
  ***ip nat inside source list 1 interface virtual-template 1 overload oer***
  NAT awareness for SOHO. NAT session will remain in case of route change via second ISP

### Config

- ***oer border***
  Enable OER border router
- ***port <port>***
  Dynamic port used for communication between MC and BR. Must be the same on both sides
- ***local <intf>***
  Identifies source for communication with an OER MC
- ***master <ip> key-chain <name>***
  Define MC

## Phases Wheel

### Learn (BR)

- The list of traffic classes entries is calles 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)
- BR profiles interesting traffic which has to be optimized by learning flows that pass through a router. Non-interfesting traffic is ignored
- BR sorts traffic based on delay and throughput and sends it to MC
- Next hops on each border router cannot be from the same subnet (exchange points)

### Measure (BR)

- PfR automatically configures (virtualy) IP SLA ICMP probes and NetFlow configurations. No explicit NetFlow or IP SLAs configuration is required
- OER measures the performance of traffic classes using active and passive monitoring techniques but it also measures, by default, the utilization of links
- Active monitoring generates synthetic traffic to emulate the traffic class that is being monitored
- Passive monitoring measures metrics of the traffic flow traversing the device in the data path
- By default all traffic classes are passively monitored using integrated NetFlow functionality and out-of-policy traffic classes are actively monitored using IP SLA functionality (learned probe)

### Apply Policy (MC)

- If multiple exists exist including existing one, use existing one, otherwise randomly pick exit
- OER compares the results with a set of configured low and high thresholds for each metric
- policies define the criteria for determining an Oot-Of-Profile event.
- Can be applied globaly, per traffic (learned automaticaly or defined manualy) class and per external link (overwrites previous)
- By default, OER runs in an observe mode during the profile, measure, and apply policy phases (no changes to network are made untill OER is configured to controll the traffic)
- Every rule has three attributes: scope (traffic class), action (insert a route), and condition that triggers the rule (acceptable thresholds)

### Enforce (BR)

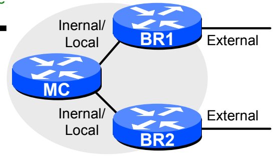
- Routing can be manipulated with artificialy injected more-specific routes. Measured prefixes' parent route (the same or wider prefix) with a valid next hop must exist for prefix to be injected
- In control mode commands are sent back to the border routers to alter routing in the OER managed network to implement the policy decisions
- If an IGP is deployed in your network, static route redistribution must be configured
- OER initiates route changes when one of the following occurs: traffic class goes OOP, exit link goes OOP or periodic timer expires and the select exit mode is configured as select best mode

### Verify (MC)

- After the controls are introduced, OER will verify that the optimized traffic is flowing through the preferred exit or entrance links at the network edge

## Interfaces

- Local interfaces – used for communication beween MC and BRs. loopback interface should be configured if MC and BR are on the same router. Configured only on BR
- Internal interfaces - used only for passive performance monitoring with NetFlow. NetFlow configuration is not required. Internal interfaces do not forward traffic
- External interfaces - OER-managed exit links to forward traffic. At least two for OER-managed domain, at leas one on each BR

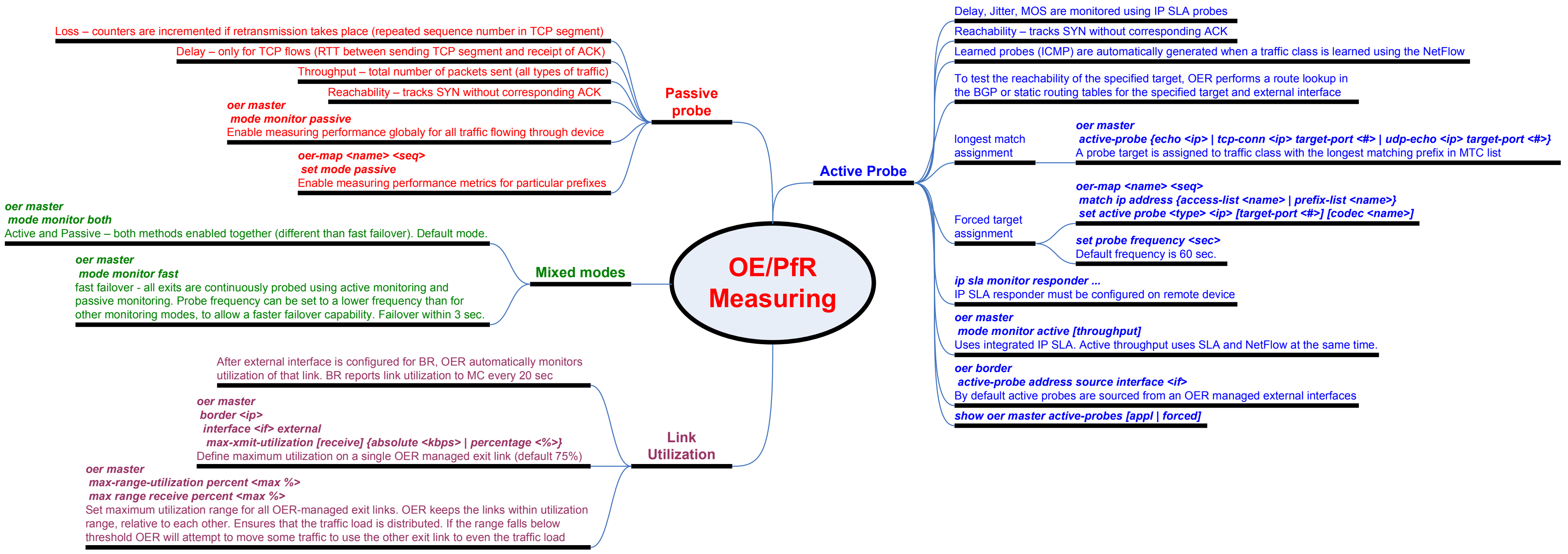


## Authentication

- ***key chain <name>***
  ***key <id>***
  ***key-string <text>***
  Key-ID and key-sting must match on MC and BR
- Authentication is required. MD5 key-chain **must be** configured between MC and BRs, even if they are configured on the same router

## Verify

- ***show oer {master | border}***
- ***show oer master traffic-class***
- ***show oer master prefix <prefix> policy***
- ***show oer border passive learn***
- ***show ip cache verbose flow***
- ***show oer border passive cache {learned | prefix} [applications]***



OE/PfR Measuring

Passive probe

Loss – counters are incremented if retransmission takes place (repeated sequence number in TCP segment)

Delay – only for TCP flows (RTT between sending TCP segment and receipt of ACK)

Throughput – total number of packets sent (all types of traffic)

Reachability – tracks SYN without corresponding ACK

***oer master***
***mode monitor passive***
Enable measuring performance globaly for all traffic flowing through device

***oer-map <name> <seq>***
***set mode passive***
Enable measuring performance metrics for particular prefixes

Mixed modes

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

***oer master***
***mode monitor fast***
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.

Link Utilization

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

***oer master***
***border <ip>***
***interface <if> external***
***max-xmit-utilization [receive] {absolute <kbps> | percentage <%>}***
Define maximum utilization on a single OER managed exit link (default 75%)

***oer master***
***max-range-utilization percent <max %>***
***max range receive percent <max %>***
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

Active Probe

Delay, Jitter, MOS are monitored using IP SLA probes

Reachability – tracks SYN without corresponding 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

longest match assignment

***oer master***
***active-probe {echo <ip> | tcp-conn <ip> target-port <#> | udp-echo <ip> target-port <#>}***
A probe target is assigned to traffic class with the longest matching prefix in MTC list

Forced target assignment

***oer-map <name> <seq>***
***match ip address {access-list <name> | prefix-list <name>}***
***set active probe <type> <ip> [target-port <#>] [codec <name>]***

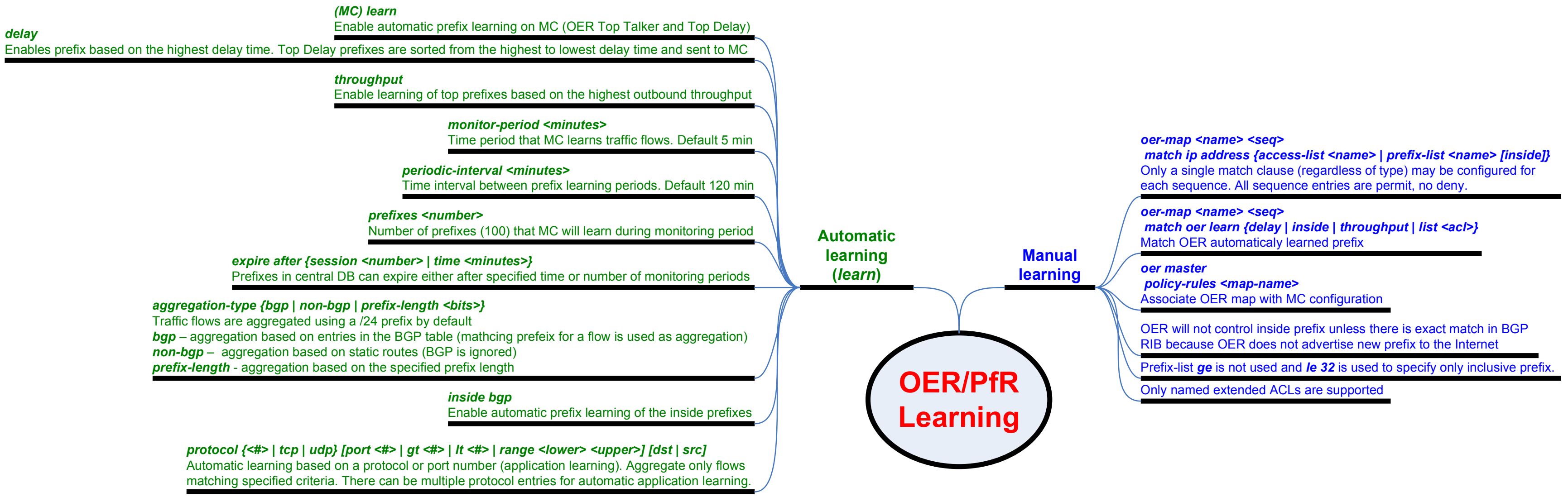
***set probe frequency <sec>***
Default frequency is 60 sec.

***ip sla monitor responder ...***
IP SLA responder must be configured on remote device

***oer master***
***mode monitor active [throughput]***
Uses integrated IP SLA. Active throughput uses SLA and NetFlow at the same time.

***oer border***
***active-probe address source interface <if>***
By default active probes are sourced from an OER managed external interfaces

***show oer master active-probes [appl | forced]***



OER/PfR Learning

Automatic learning (***learn***)

***(MC) learn***
Enable automatic prefix learning on MC (OER Top Talker and Top Delay)

***delay***
Enables prefix based on the highest delay time. Top Delay prefixes are sorted from the highest to lowest delay time and sent to MC

***throughput***
Enable learning of top prefixes based on the highest outbound throughput

***monitor-period <minutes>***
Time period that MC learns traffic flows. Default 5 min

***periodic-interval <minutes>***
Time interval between prefix learning periods. Default 120 min

***prefixes <number>***
Number of prefixes (100) that MC will learn during monitoring period

***expire after {session <number> | time <minutes>}***
Prefixes in central DB can expire either after specified time or number of monitoring periods

***aggregation-type {bgp | non-bgp | prefix-length <bits>}***
Traffic flows are aggregated using a /24 prefix by default
***bgp*** – aggregation based on entries in the BGP table (mathcing prefeix for a flow is used as aggregation)
***non-bgp*** – aggregation based on static routes (BGP is ignored)
***prefix-length*** - aggregation based on the specified prefix length

***inside bgp***
Enable automatic prefix learning of the inside prefixes

***protocol {<#> | tcp | udp} [port <#> | gt <#> | lt <#> | range <lower> <upper>] [dst | src]***
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.

Manual learning

***oer-map <name> <seq>***
***match ip address {access-list <name> | prefix-list <name> [inside]}***
Only a single match clause (regardless of type) may be configured for each sequence. All sequence entries are permit, no deny.

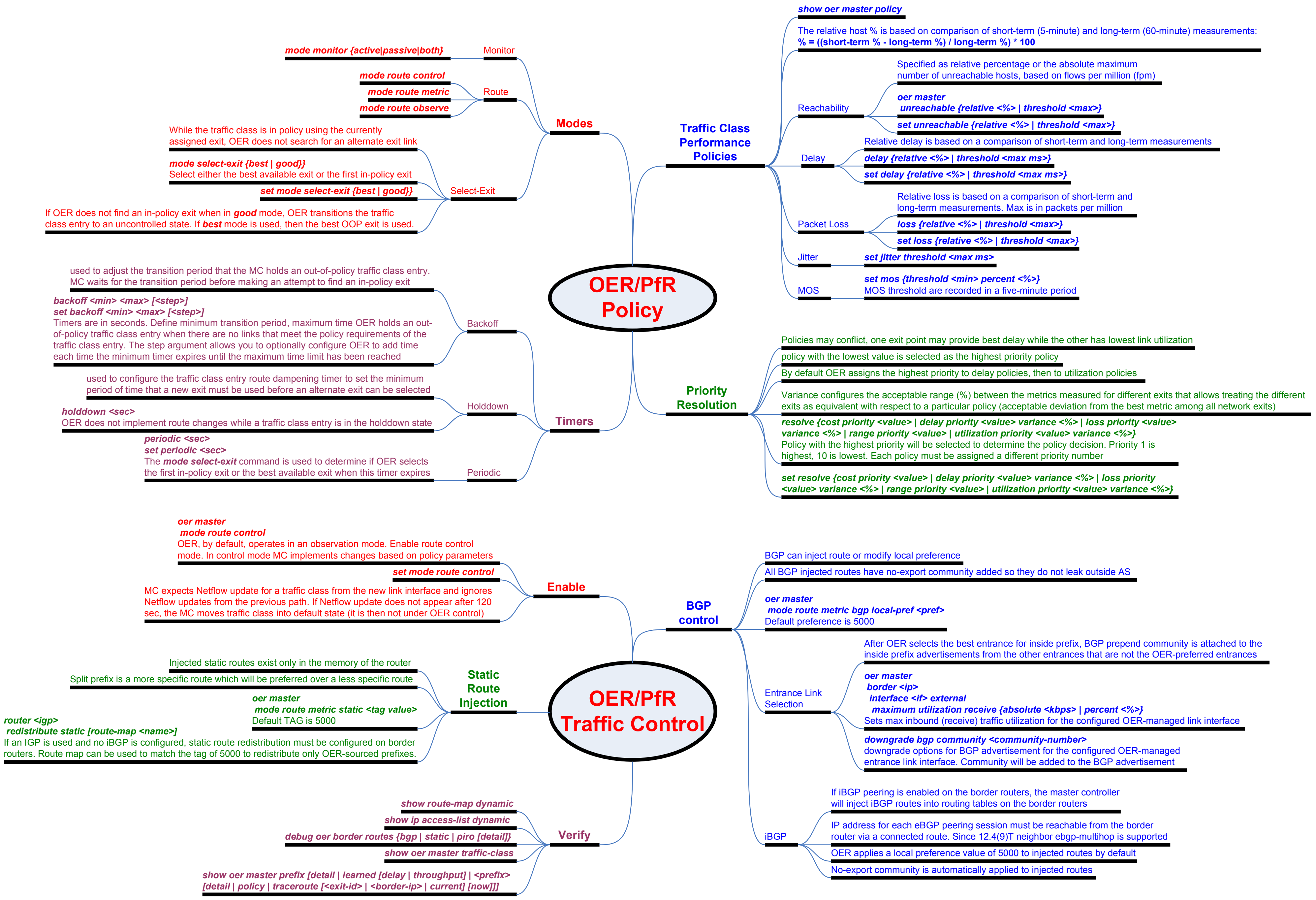
***oer-map <name> <seq>***
***match oer learn {delay | inside | throughput | list <acl>}***
Match OER automaticaly learned prefix

***oer master***
***policy-rules <map-name>***
Associate OER map with MC configuration

OER will not control inside prefix unless there is exact match in BGP RIB because OER does not advertise new prefix to the Internet

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

Only named extended ACLs are supported



# OER/PfR Policy

## Modes

- Monitor
  - *mode monitor {active|passive|both}*
- Route
  - *mode route control*
  - *mode route metric*
  - *mode route observe*
- Select-Exit
  - While the traffic class is in policy using the currently assigned exit, OER does not search for an alternate exit link
  - *mode select-exit {best | good}}*
    Select either the best available exit or the first in-policy exit
  - *set mode select-exit {best | good}}*
  - If OER does not find an in-policy exit when in ***good*** mode, OER transitions the traffic class entry to an uncontrolled state. If ***best*** mode is used, then the best OOP exit is used.

## Timers

- Backoff
  - used to adjust the transition period that the MC holds an out-of-policy traffic class entry. MC waits for the transition period before making an attempt to find an in-policy exit
  - *backoff <min> <max> [<step>]*
    *set backoff <min> <max> [<step>]*
    Timers are in seconds. Define minimum transition period, maximum time OER holds an out-of-policy traffic class entry when there are no links that meet the policy requirements of the traffic class entry. The step argument allows you to optionally configure OER to add time each time the minimum timer expires until the maximum time limit has been reached
- Holddown
  - used to configure the traffic class entry route dampening timer to set the minimum period of time that a new exit must be used before an alternate exit can be selected
  - *holddown <sec>*
    OER does not implement route changes while a traffic class entry is in the holddown state
- Periodic
  - *periodic <sec>*
    *set periodic <sec>*
    The ***mode select-exit*** command is used to determine if OER selects the first in-policy exit or the best available exit when this timer expires

## Traffic Class Performance Policies

- ***show oer master policy***
- The relative host % is based on comparison of short-term (5-minute) and long-term (60-minute) measurements:
  **% = ((short-term % - long-term %) / long-term %) * 100**
- Reachability
  - Specified as relative percentage or the absolute maximum number of unreachable hosts, based on flows per million (fpm)
  - ***oer master***
    ***unreachable {relative <%> | threshold <max>}***
  - ***set unreachable {relative <%> | threshold <max>}***
- Delay
  - Relative delay is based on a comparison of short-term and long-term measurements
  - ***delay {relative <%> | threshold <max ms>}***
  - ***set delay {relative <%> | threshold <max ms>}***
- Packet Loss
  - Relative loss is based on a comparison of short-term and long-term measurements. Max is in packets per million
  - ***loss {relative <%> | threshold <max>}***
  - ***set loss {relative <%> | threshold <max>}***
- Jitter
  - ***set jitter threshold <max ms>***
- MOS
  - ***set mos {threshold <min> percent <%>}***
    MOS threshold are recorded in a five-minute period

## Priority Resolution

- Policies may conflict, one exit point may provide best delay while the other has lowest link utilization
- policy with the lowest value is selected as the highest priority policy
- By default OER assigns the highest priority to delay policies, then to utilization policies
- Variance configures the acceptable range (%) between the metrics measured for different exits that allows treating the different exits as equivalent with respect to a particular policy (acceptable deviation from the best metric among all network exits)
- ***resolve {cost priority <value> | delay priority <value> variance <%> | loss priority <value> variance <%> | range priority <value> | utilization priority <value> variance <%>}***
  Policy with the highest priority will be selected to determine the policy decision. Priority 1 is highest, 10 is lowest. Each policy must be assigned a different priority number
- ***set resolve {cost priority <value> | delay priority <value> variance <%> | loss priority <value> variance <%> | range priority <value> | utilization priority <value> variance <%>}***

# OER/PfR Traffic Control

## Enable

- ***oer master***
  ***mode route control***
  OER, by default, operates in an observation mode. Enable route control mode. In control mode MC implements changes based on policy parameters
- ***set mode route control***
- MC expects Netflow update for a traffic class from the new link interface and ignores Netflow updates from the previous path. If Netflow update does not appear after 120 sec, the MC moves traffic class into default state (it is then not under OER control)

## Static Route Injection

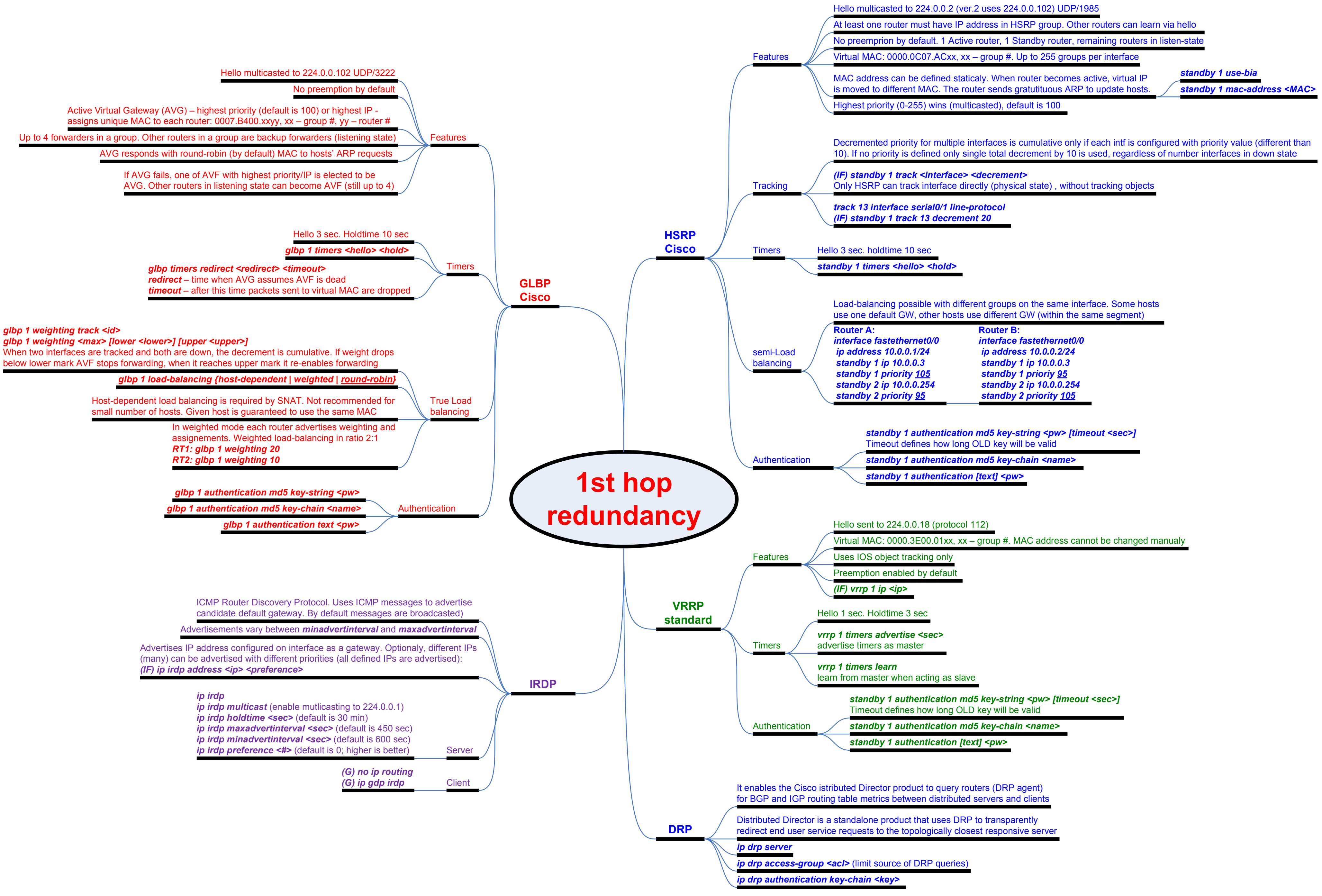
- Injected static routes exist only in the memory of the router
- Split prefix is a more specific route which will be preferred over a less specific route
- ***oer master***
  ***mode route metric static <tag value>***
  Default TAG is 5000
- ***router <igp>***
  ***redistribute static [route-map <name>]***
  If an IGP is used and no iBGP is configured, static route redistribution must be configured on border routers. Route map can be used to match the tag of 5000 to redistribute only OER-sourced prefixes.

## Verify

- ***show route-map dynamic***
- ***show ip access-list dynamic***
- ***debug oer border routes {bgp | static | piro [detail]}***
- ***show oer master traffic-class***
- ***show oer master prefix [detail | learned [delay | throughput] | <prefix> [detail | policy | traceroute [<exit-id> | <border-ip> | current] [now]]]***

## BGP control

- BGP can inject route or modify local preference
- All BGP injected routes have no-export community added so they do not leak outside AS
- ***oer master***
  ***mode route metric bgp local-pref <pref>***
  Default preference is 5000
- Entrance Link Selection
  - After OER selects the best entrance for inside prefix, BGP prepend community is attached to the inside prefix advertisements from the other entrances that are not the OER-preferred entrances
  - ***oer master***
    ***border <ip>***
    ***interface <if> external***
    ***maximum utilization receive {absolute <kbps> | percent <%>}***
    Sets max inbound (receive) traffic utilization for the configured OER-managed link interface
  - ***downgrade bgp community <community-number>***
    downgrade options for BGP advertisement for the configured OER-managed entrance link interface. Community will be added to the BGP advertisement
- iBGP
  - If iBGP peering is enabled on the border routers, the master controller will inject iBGP routes into routing tables on the border routers
  - IP address for each eBGP peering session must be reachable from the border router via a connected route. Since 12.4(9)T neighbor ebgp-multihop is supported
  - OER applies a local preference value of 5000 to injected routes by default
  - No-export community is automatically applied to injected routes



# 1st hop redundancy

## GLBP Cisco

### Features
- Hello multicasted to 224.0.0.102 UDP/3222
- No preemption by default
- Active Virtual Gateway (AVG) – highest priority (default is 100) or highest IP - assigns unique MAC to each router: 0007.B400.xxyy, xx – group #, yy – router #
- Up to 4 forwarders in a group. Other routers in a group are backup forwarders (listening state)
- AVG responds with round-robin (by default) MAC to hosts' ARP requests
- If AVG fails, one of AVF with highest priority/IP is elected to be AVG. Other routers in listening state can become AVF (still up to 4)

### Timers
- Hello 3 sec. Holdtime 10 sec
- ***glbp 1 timers <hello> <hold>***
- ***glbp timers redirect <redirect> <timeout>***
  ***redirect*** – time when AVG assumes AVF is dead
  ***timeout*** – after this time packets sent to virtual MAC are dropped

### True Load balancing
- ***glbp 1 weighting track <id>***
  ***glbp 1 weighting <max> [lower <lower>] [upper <upper>]***
  When two interfaces are tracked and both are down, the decrement is cumulative. If weight drops below lower mark AVF stops forwarding, when it reaches upper mark it re-enables forwarding
- ***glbp 1 load-balancing {host-dependent | weighted | round-robin}***
- Host-dependent load balancing is required by SNAT. Not recommended for small number of hosts. Given host is guaranteed to use the same MAC
- In weighted mode each router advertises weighting and assignements. Weighted load-balancing in ratio 2:1
  ***RT1: glbp 1 weighting 20***
  ***RT2: glbp 1 weighting 10***

### Authentication
- ***glbp 1 authentication md5 key-string <pw>***
- ***glbp 1 authentication md5 key-chain <name>***
- ***glbp 1 authentication text <pw>***

## HSRP Cisco

### Features
- Hello multicasted to 224.0.0.2 (ver.2 uses 224.0.0.102) UDP/1985
- At least one router must have IP address in HSRP group. Other routers can learn via hello
- No preemprion by default. 1 Active router, 1 Standby router, remaining routers in listen-state
- Virtual MAC: 0000.0C07.ACxx, xx – group #. Up to 255 groups per interface
- MAC address can be defined staticaly. When router becomes active, virtual IP is moved to different MAC. The router sends gratutituous ARP to update hosts.
  - ***standby 1 use-bia***
  - ***standby 1 mac-address <MAC>***
- Highest priority (0-255) wins (multicasted), default is 100

### Tracking
- Decremented priority for multiple interfaces is cumulative only if each intf is configured with priority value (different than 10). If no priority is defined only single total decrement by 10 is used, regardless of number interfaces in down state
- ***(IF) standby 1 track <interface> <decrement>***
  Only HSRP can track interface directly (physical state) , without tracking objects
- ***track 13 interface serial0/1 line-protocol***
  ***(IF) standby 1 track 13 decrement 20***

### Timers
- Hello 3 sec. holdtime 10 sec
- ***standby 1 timers <hello> <hold>***

### semi-Load balancing
- Load-balancing possible with different groups on the same interface. Some hosts use one default GW, other hosts use different GW (within the same segment)
- Router A:

```
interface fastethernet0/0
 ip address 10.0.0.1/24
 standby 1 ip 10.0.0.3
 standby 1 priority 105
 standby 2 ip 10.0.0.254
 standby 2 priority 95
```

- Router B:

```
interface fastethernet0/0
 ip address 10.0.0.2/24
 standby 1 ip 10.0.0.3
 standby 1 prioriy 95
 standby 2 ip 10.0.0.254
 standby 2 priority 105
```

### Authentication
- ***standby 1 authentication md5 key-string <pw> [timeout <sec>]***
  Timeout defines how long OLD key will be valid
- ***standby 1 authentication md5 key-chain <name>***
- ***standby 1 authentication [text] <pw>***

## VRRP standard

### Features
- Hello sent to 224.0.0.18 (protocol 112)
- Virtual MAC: 0000.3E00.01xx, xx – group #. MAC address cannot be changed manualy
- Uses IOS object tracking only
- Preemption enabled by default
- ***(IF) vrrp 1 ip <ip>***

### Timers
- Hello 1 sec. Holdtime 3 sec
- ***vrrp 1 timers advertise <sec>***
  advertise timers as master
- ***vrrp 1 timers learn***
  learn from master when acting as slave

### Authentication
- ***standby 1 authentication md5 key-string <pw> [timeout <sec>]***
  Timeout defines how long OLD key will be valid
- ***standby 1 authentication md5 key-chain <name>***
- ***standby 1 authentication [text] <pw>***

## IRDP
- ICMP Router Discovery Protocol. Uses ICMP messages to advertise candidate default gateway. By default messages are broadcasted)
- Advertisements vary between ***minadvertinterval*** and ***maxadvertinterval***
- Advertises IP address configured on interface as a gateway. Optionaly, different IPs (many) can be advertised with different priorities (all defined IPs are advertised):
  ***(IF) ip irdp address <ip> <preference>***

### Server
- ***ip irdp***
  ***ip irdp multicast*** (enable mutlicasting to 224.0.0.1)
  ***ip irdp holdtime <sec>*** (default is 30 min)
  ***ip irdp maxadvertinterval <sec>*** (default is 450 sec)
  ***ip irdp minadvertinterval <sec>*** (default is 600 sec)
  ***ip irdp preference <#>*** (default is 0; higher is better)

### Client
- ***(G) no ip routing***
  ***(G) ip gdp irdp***

## DRP
- It enables the Cisco istributed Director product to query routers (DRP agent) for BGP and IGP routing table metrics between distributed servers and clients
- Distributed Director is a standalone product that uses DRP to transparently redirect end user service requests to the topologically closest responsive server
- ***ip drp server***
- ***ip drp access-group <acl>*** (limit source of DRP queries)
- ***ip drp authentication key-chain <key>***

# NAT

## Features

- **Inside local** – how inside address is seen localy (by inside hosts)
- **Inside global** – how inside address is seen globaly (by outside hosts)
- **Outside local** – how outside address is seen localy (by inside hosts)
- **Outside global** – how outside address is seen globaly (by outside hosts)
- Not supported: Routing table updates, DNS zone transfers, BOOTP, SNMP

### Fragments

- NAT keeps stateful information about fragments. If a first fragment is translated, information is kept so that subsequent fragments are translated the same way.
- If a fragment arrives before the first fragment, the NAT holds the fragment until the first fragment arrives

### FTP Pasive

- PORT and PASV commands carry IP addresses in ASCII form
- When the address is translated, the message size can change. If the size message remains the same, the Cisco NAT recalculates only the TCP checksum
- If the translation results in a smaller message, the NAT pads the message with ACSII zeros to make it the same size as the original message
- TCP SEQ and ACK numbers are based directly on the length of the TCP segments. NAT tracks changes in SEQ and ACK numbers. It takes place if translated message is larger than original one

## Multihoming to 2 ISPs

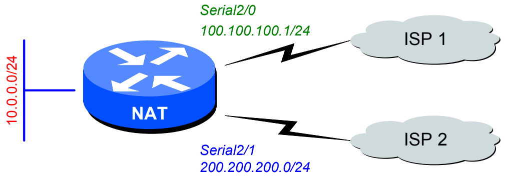
```
ip nat pool ISP1 100.100.100.10 100.100.100.50 prefix-length 24
ip nat inside source route-map ISP1_MAP pool ISP1

ip nat pool ISP2 200.200.200.10 200.200.200.50 prefix-length 24
ip nat inside source route-map ISP2_MAP pool ISP2

route-map ISP1_MAP permit 10
 match ip address 1
 match interface Serial2/0 ! outgoing interface

route-map ISP2_MAP permit 10
 match ip address 1
 match interface Serial2/1 ! outgoing interface

access-list 1 permit 10.0.0.0 0.0.0.255
```



If inside host opens route-map (only) based dynamic translation, outside host can be also able to initiate connection to inside host (bi-directional traffic initiation is allowed for specific one-to-one mapping, which is created in addition to extendable mapping)
***ip nat inside source route-map ISP2_MAP pool ISP2 reversible***

## Static

- ***ip nat inside source static <inside local> <inside global>***
- Static NAT (for 1:1 IP address) performs tranlsations in both directions. Packets initiated from outside into inside are translated, but also packets initiated from inside to outside are translated.
- Network translation assignes last octed one-to-one
  ***ip nat inside source static network <local net> <global net> /24***
- Statically mapping an IG address to more than one IL address is normally not allowed. To allow service distribution an ***extendable*** keyword must be used. However, this is only for incoming traffic from outside. Outgoing traffic (initiated from inside) falls under dynamic NAT. If it's not configured, traffic is dropped.
  ***ip nat inside source static tcp 192.168.1.1 21 199.198.5.1 21 extendable***
  ***ip nat inside source static tcp 192.168.1.3 80 199.198.5.1 80 extendable***
- By default IG address is added to local IP aliases (show ip alias), so the router can terminate traffic (other than NATed) on itself, using this IP. If no-alias keyword is used, IG address is not added to aliases. Router will not terminate the traffic, but it will respond to ARP requests.
  ***ip nat inside source static tcp 192.168.1.1 21 199.198.5.1 21 no-alias***

### With HSRP

- Active router is the only one which is performing NAT translation
- R1/R2:

```
interface <if>
 standby name <HSRP name>
 ip nat inside
```

- ***ip nat inside source static <IL> <IG> redundancy <name>***

## Load balancing

- In NAT TCP load balancing, non-TCP packets pass through the NAT untranslated
- **1**. Define local servers IL addresses:
  ***ip nat pool <name> <start> <end> prefix-length <bits> type rotary***
  or using more flexible way:

```
ip nat pool <name> prefix-length <bits> type rotary
 address <start1> <end1>
 address <start2> <end2>
```

- **2.** Associate global IP (single IPs), by which local servers are seen from outside
  ***ip nat inside destination list <acl> pool <name>***
  ***access-list <acl> permit <global IP>***
- ***ip alias <global IP> <port>***
  It may be required to create an IP alias for global IP, so the router accepts traffic for that IP it extended ACL is used with specific port numbers. The IP alias is not automaticaly created by the NAT

## Dynamic

### PAT

- ***ip nat inside source list 1 interface Serial0 overload***
  All inside sources are translated to single interface IP address. Up to 65535 IL addresses could theoretically be mapped to a single IG address (based on the 16-bit port number)
- Each NAT entry uses approximately 160 bytes of memory, so 65535 entries would consume more than 10 MB of memory and large amounts of CPU power

- ***ip nat pool <name> <start> <end> netmask <mask> [type match-host]***
  Host portion of the IG address will match the host portion of the IL address. The netmask portion of the commands acts as a sanity check, ensuring that such addresses as 204.15.87.255 are not mapped
- ***ip nat inside source list <acl> pool <name>***
  Translate dynamicaly source addresses of inside hosts

## Stateful

### With HSRP

R1/R2:

```
interface <if>
 standaby name <HSRP name>
 standaby ...
 ip nat inside
```

R1/R2:

```
ip nat stateful id <id>
 redundancy <HSRP name>
  mapping-id <id>
```

- Stateful-id must be unique for each router
- Mapping-id identifies translations and must be the same on both routers

***ip nat inside source list <acl> pool <name> mapping <mapping id>***

### No HSRP

R1:

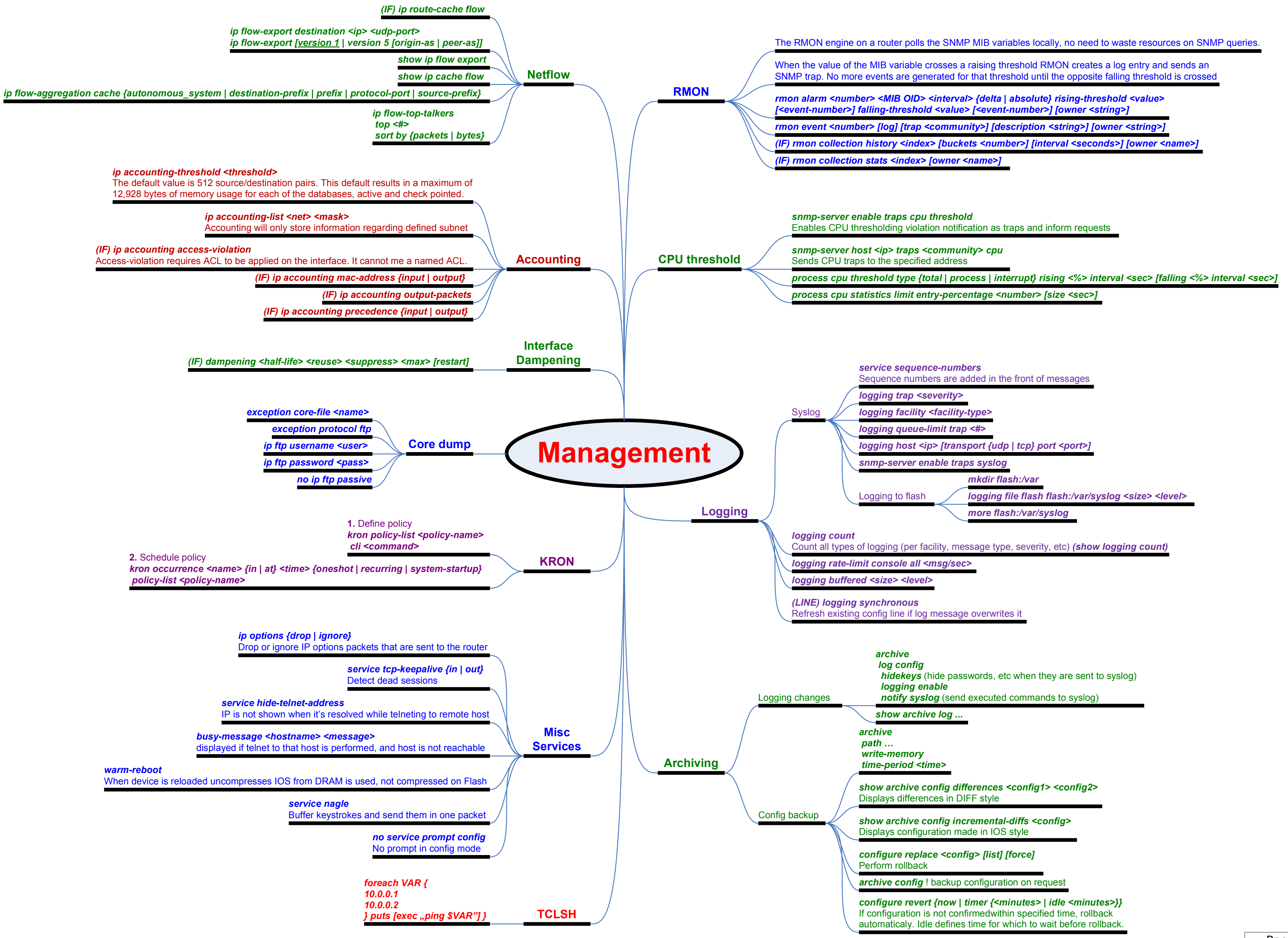
```
ip nat stateful id <id>
 primary <R1 IP>
  peer <R2 IP>
   mapping-id <id>
```

R2:

```
ip nat stateful id <id>
 backup <R2 IP>
  peer <R1 IP>
   mapping-id <id>
```

***show ip snat peer <ip>***
Show translations on peer router

***show ip snat distributed verbose***



# Management

## Netflow

***(IF) ip route-cache flow***

***ip flow-export destination <ip> <udp-port>***
***ip flow-export [version 1 | version 5 [origin-as | peer-as]]***

***show ip flow export***

***show ip cache flow***

***ip flow-aggregation cache {autonomous_system | destination-prefix | prefix | protocol-port | source-prefix}***

***ip flow-top-talkers***
***top <#>***
***sort by {packets | bytes}***

## Accounting

***ip accounting-threshold <threshold>***
The default value is 512 source/destination pairs. This default results in a maximum of 12,928 bytes of memory usage for each of the databases, active and check pointed.

***ip accounting-list <net> <mask>***
Accounting will only store information regarding defined subnet

***(IF) ip accounting access-violation***
Access-violation requires ACL to be applied on the interface. It cannot me a named ACL.

***(IF) ip accounting mac-address {input | output}***

***(IF) ip accounting output-packets***

***(IF) ip accounting precedence {input | output}***

## Interface Dampening

***(IF) dampening <half-life> <reuse> <suppress> <max> [restart]***

## Core dump

***exception core-file <name>***

***exception protocol ftp***

***ip ftp username <user>***

***ip ftp password <pass>***

***no ip ftp passive***

## KRON

**1.** Define policy
***kron policy-list <policy-name>***
***cli <command>***

**2.** Schedule policy
***kron occurrence <name> {in | at} <time> {oneshot | recurring | system-startup}***
***policy-list <policy-name>***

## Misc Services

***ip options {drop | ignore}***
Drop or ignore IP options packets that are sent to the router

***service tcp-keepalive {in | out}***
Detect dead sessions

***service hide-telnet-address***
IP is not shown when it's resolved while telneting to remote host

***busy-message <hostname> <message>***
displayed if telnet to that host is performed, and host is not reachable

***warm-reboot***
When device is reloaded uncompresses IOS from DRAM is used, not compressed on Flash

***service nagle***
Buffer keystrokes and send them in one packet

***no service prompt config***
No prompt in config mode

## TCLSH

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

## RMON

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>]***

***(IF) rmon collection history <index> [buckets <number>] [interval <seconds>] [owner <name>]***

***(IF) rmon collection stats <index> [owner <name>]***

## CPU threshold

***snmp-server enable traps cpu threshold***
Enables CPU thresholding violation notification as traps and inform requests

***snmp-server host <ip> traps <community> cpu***
Sends CPU traps to the specified address

***process cpu threshold type {total | process | interrupt} rising <%> interval <sec> [falling <%> interval <sec>]***

***process cpu statistics limit entry-percentage <number> [size <sec>]***

## Logging

### Syslog

***service sequence-numbers***
Sequence numbers are added in the front of messages

***logging trap <severity>***

***logging facility <facility-type>***

***logging queue-limit trap <#>***

***logging host <ip> [transport {udp | tcp} port <port>]***

***snmp-server enable traps syslog***

Logging to flash

- ***mkdir flash:/var***
- ***logging file flash flash:/var/syslog <size> <level>***
- ***more flash:/var/syslog***

***logging count***
Count all types of logging (per facility, message type, severity, etc) ***(show logging count)***

***logging rate-limit console all <msg/sec>***

***logging buffered <size> <level>***

***(LINE) logging synchronous***
Refresh existing config line if log message overwrites it

## Archiving

### Logging changes

***archive***
***log config***
***hidekeys*** (hide passwords, etc when they are sent to syslog)
***logging enable***
***notify syslog*** (send executed commands to syslog)

***show archive log ...***

### Config backup

***archive***
***path …***
***write-memory***
***time-period <time>***

***show archive config differences <config1> <config2>***
Displays differences in DIFF style

***show archive config incremental-diffs <config>***
Displays configuration made in IOS style

***configure replace <config> [list] [force]***
Perform rollback

***archive config*** ! backup configuration on request

***configure revert {now | timer {<minutes> | idle <minutes>}}***
If configuration is not confirmedwithin specified time, rollback automaticaly. Idle defines time for which to wait before rollback.

# DNS

## Authoritative server

- ***ip dns server***
- ***ip dns primary <domain> soa <ns> <email>***
- ***ip host <domain> ns <ip>***
- ***ip host <fqdn> <ip1> ... <ip6>***
- ***ip domain round-robin***
- ***ip domain {timeout <sec> | retry <#>}***

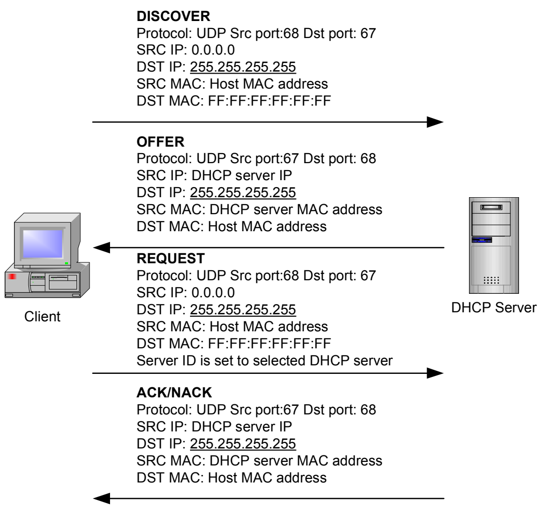
## Caching server/Client

- ***ip name-server <ip>***
- ***ip dns server***
- ***ip domain lookup***

## Spoofing

- ***ip dns server***
- ***ip name-server <ip>***
- ***ip dns spoofing [<ip>]***
  If upstream DNS server is up, router will proxy and forward queries. If upstream is down, router will respond to all queries with pre-configured IP only if query is not for router's own interface, then it replies with interface IP on which query was received.

| OP Code | HW Type | HW Len | Hop count |
|---|---|---|---|
| Transaction ID (32b) | | | |
| Seconds (16b) | | Flags (16b) | |
| Client IP Address (CIADDR) (32b) | | | |
| Your IP Address (YIADDR) (32b) | | | |
| Server IP Address (SIADDR) (32b) | | | |
| Gateway IP Address (GIADDR) (32b) | | | |
| Client HW Address (CHADDR) (16b) | | | |
| Server name (SNAME) (64b) | | | |
| Boot filename  (128b) | | | |
| DHCP options | | | |



**DISCOVER**
Protocol: UDP Src port:68 Dst port: 67
SRC IP: 0.0.0.0
DST IP: 255.255.255.255
SRC MAC: Host MAC address
DST MAC: FF:FF:FF:FF:FF:FF

**OFFER**
Protocol: UDP Src port:67 Dst port: 68
SRC IP: DHCP server IP
DST IP: 255.255.255.255
SRC MAC: DHCP server MAC address
DST MAC: Host MAC address

**REQUEST**
Protocol: UDP Src port:68 Dst port: 67
SRC IP: 0.0.0.0
DST IP: 255.255.255.255
SRC MAC: Host MAC address
DST MAC: FF:FF:FF:FF:FF:FF
Server ID is set to selected DHCP server

**ACK/NACK**
Protocol: UDP Src port:67 Dst port: 68
SRC IP: DHCP server IP
DST IP: 255.255.255.255
SRC MAC: DHCP server MAC address
DST MAC: Host MAC address

# DHCP

## Features

- UDP/67 server; UDP/68 client
- based on the Bootstrap Protocol (BOOTP)
- Server responding to client's Discover and Request messages also uses broadcast to inform other possible DHCP server on a LAN, that the request has been served
- Address is assigned with lease time. Client can extend lease time dynamically
- ***service dhcp*** (enabled by default)

## Proxy

When a dialing client requests an IP address via IPCP, the dialed router can request this IP on client's behalf from remote DHCP server, acting as a proxy. The dialed router uses own IP from PPP interface to set *giaddr* in the request

```
interface <if>
 ip address <ip> <mask>
 encapsulation ppp
 peer default ip address dhcp
```

```
ip address-pool dhcp-proxy-client
ip dhcp-server <ip>
```

## Relay

- ***ip helper address <ip> [redundancy <HSRP name>]***
  Broadcast is changed to directed unicast with router's LAN interface's IP address as a source. This feature is used if DHCP server is not on the same segment as clients (broadcast is not propagated through a router). If redundancy is used, only active router will forward queries to the server
- If a client is in local network *giaddr* in HDCP DISCOVER message is set to 0 (zero), and a pool is choosen from interface on which the message was received. If ***ip helper address*** is used, *giaddr* is set to forwarding router interface's IP, and a pool is choosed from this particular IP regardless of interface on which unicasted request was received..
- ***(G) ip dhcp smart-relay***
  Relay agent attempts to forward the primary address as the gateway address three times. If no response is received then secondary addresses on relay agtent's interface are used.

## On-demand pool

This feature is usefull when WAN links get's all IP information dynamicaly assigned, and DHCP options (DNS, domain, etc) need to be passed to clients behind a router.

R2 PE:
```
interface <if>
 encapsulation ppp
 ip address <ip> <mask>
 peer default ip address <peer-ip>
 ppp ipcp mask <mask>
 ppp ipcp dns <dns1> <dns2>
 no peer neighbor-route
```

R1 CPE:
```
interface <if>
 encapsulation ppp
 ip address negotiated
 ppp ipcp netmask request
 ppp ipcp dns request
```

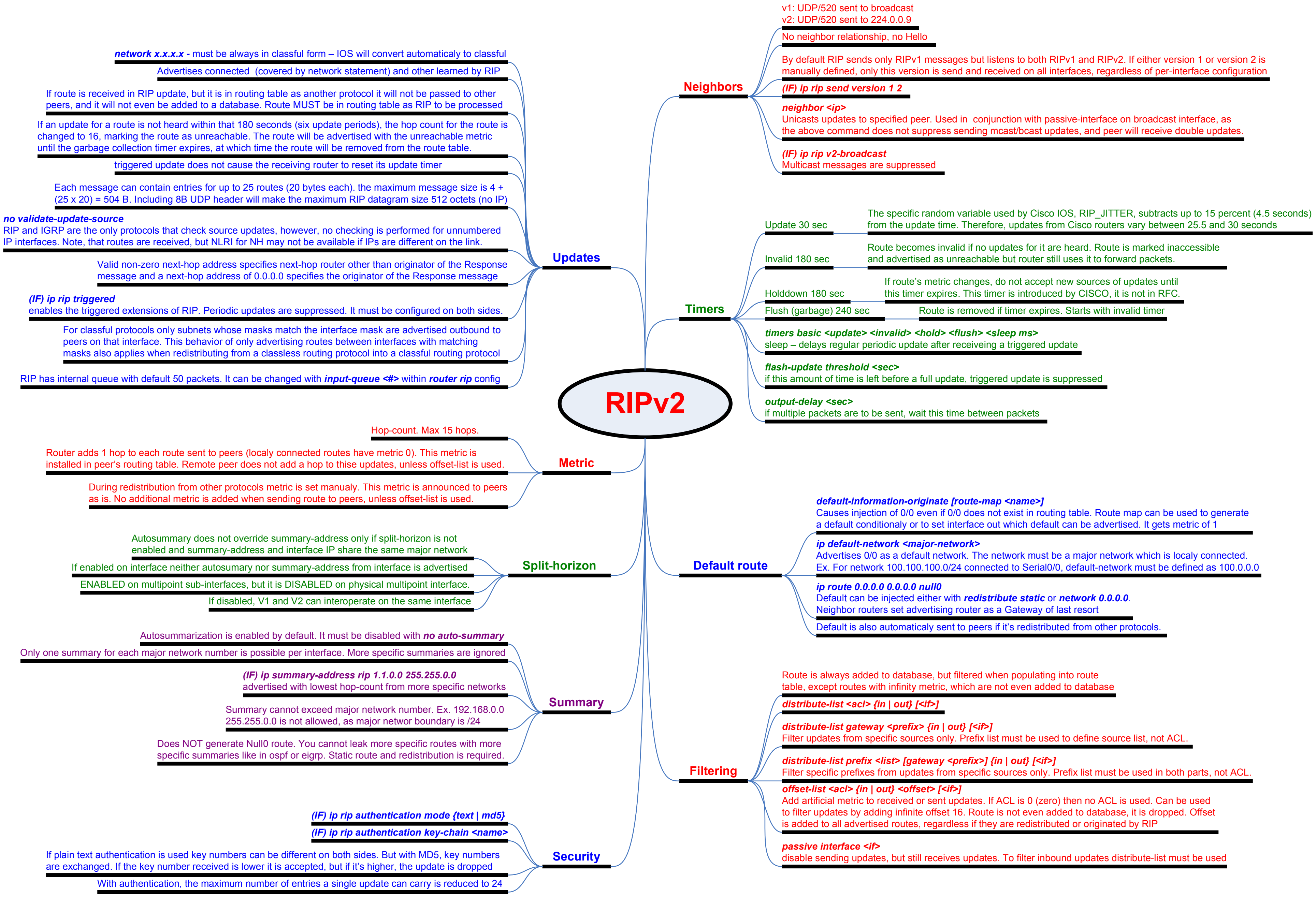
```
ip dhcp pool <name>
 import all
 origin ipcp
```

## Client

- ***release dhcp <if>***
  Force interface to release and renew IPD address
- ***(IF) ip dhcp client request ...***
  Request additional parameters (options)
- ***(IF) ip dhcp client lease <deys> [<hours>]***
  Request specific lease time for an address
- ***(IF) ip dhcp client client-id <if>***
  Specify Client-ID used to identify certain profile on DHCP server
- ***(IF) ip address dhcp***
  configure interface IP from DHCP

## Server

- ***ip dhcp pool <name>***
  ***network <net> <mask>***
  ***dns-server <ip>***
  ***domain-name <name>***
  ***lease <days> [<hours>]***
  ***option <id> <type> <value>*** (additional options – 150 TFTP server, etc)
  ***netbios-node-type <type*** (h-node Hybrid node recommended)
- ***ip dhcp exclude-address <start> <end>***
  Multiple lines defining which addresses in a network range will not be assigned to clients
- ***ip dhcp database flash:/bindings [timeout <sec>] [write-delay <sec>]***
  Configure database agent for storing bindings, and conflict logging
- ***no ip dhcp conflict-logging***
  Must be disabled if database agent is not configured (conflicts logging is possible if there is a place to store them)
- Host pools inherit entire configuration from the main pool (IP is matched against network in the pool)
  ***ip dhcp pool PC1***
  ***host <ip> /24***
  ***hardware-address <MAC>***
- When creating per-host pool, 01 must be added in the front of MAC defined as client-id (01 means ethernet media type)
- DHCP server pings IP before it is leased — ***ip dhcp ping {packets <#> | timeout <msec>}***
- ***ip dhcp bootp ignore***
  Ignore BOOTP requests sent to this DHCP server



# RIPv2

## Updates

- ***network x.x.x.x*** **-** must be always in classful form – IOS will convert automaticaly to classful
- Advertises connected  (covered by network statement) and other learned by RIP
- If route is received in RIP update, but it is in routing table as another protocol it will not be passed to other peers, and it will not even be added to a database. Route MUST be in routing table as RIP to be processed
- If an update for a route is not heard within that 180 seconds (six update periods), the hop count for the route is changed to 16, marking the route as unreachable. The route will be advertised with the unreachable metric until the garbage collection timer expires, at which time the route will be removed from the route table.
- triggered update does not cause the receiving router to reset its update timer
- Each message can contain entries for up to 25 routes (20 bytes each). the maximum message size is 4 + (25 x 20) = 504 B. Including 8B UDP header will make the maximum RIP datagram size 512 octets (no IP)
- ***no validate-update-source***
  RIP and IGRP are the only protocols that check source updates, however, no checking is performed for unnumbered IP interfaces. Note, that routes are received, but NLRI for NH may not be available if IPs are different on the link.
- Valid non-zero next-hop address specifies next-hop router other than originator of the Response message and a next-hop address of 0.0.0.0 specifies the originator of the Response message
- ***(IF) ip rip triggered***
  enables the triggered extensions of RIP. Periodic updates are suppressed. It must be configured on both sides.
- For classful protocols only subnets whose masks match the interface mask are advertised outbound to peers on that interface. This behavior of only advertising routes between interfaces with matching masks also applies when redistributing from a classless routing protocol into a classful routing protocol
- RIP has internal queue with default 50 packets. It can be changed with ***input-queue <#>*** within ***router rip*** config

## Metric

- Hop-count. Max 15 hops.
- Router adds 1 hop to each route sent to peers (localy connected routes have metric 0). This metric is installed in peer's routing table. Remote peer does not add a hop to thise updates, unless offset-list is used.
- During redistribution from other protocols metric is set manualy. This metric is announced to peers as is. No additional metric is added when sending route to peers, unless offset-list is used.

## Split-horizon

- Autosummary 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 interface neither autosumary nor summary-address from interface is advertised
- ENABLED on multipoint sub-interfaces, but it is DISABLED on physical multipoint interface.
- If disabled, V1 and V2 can interoperate on the same interface

## Summary

- Autosummarization is enabled by default. It must be disabled with ***no auto-summary***
- Only one summary for each major network number is possible per interface. More specific summaries are ignored
- ***(IF) ip summary-address rip 1.1.0.0 255.255.0.0***
  advertised with lowest hop-count from more specific networks
- Summary cannot exceed major network number. Ex. 192.168.0.0 255.255.0.0 is not allowed, as major networ boundary is /24
- Does NOT generate Null0 route. You cannot leak more specific routes with more specific summaries like in ospf or eigrp. Static route and redistribution is required.

## Security

- ***(IF) ip rip authentication mode {text | md5}***
- ***(IF) ip rip authentication key-chain <name>***
- If plain text authentication is used key numbers can be different on both sides. But with MD5, key numbers are exchanged. If the key number received is lower it is accepted, but if it's higher, the update is dropped
- With authentication, the maximum number of entries a single update can carry is reduced to 24

## Neighbors

- v1: UDP/520 sent to broadcast
  v2: UDP/520 sent to 224.0.0.9
- No neighbor relationship, no Hello
- 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 send and received on all interfaces, regardless of per-interface configuration
- ***(IF) ip rip send version 1 2***
- ***neighbor <ip>***
  Unicasts updates to specified peer. Used in  conjunction with passive-interface on broadcast interface, as the above command does not suppress sending mcast/bcast updates, and peer will receive double updates.
- ***(IF) ip rip v2-broadcast***
  Multicast messages are suppressed

## Timers

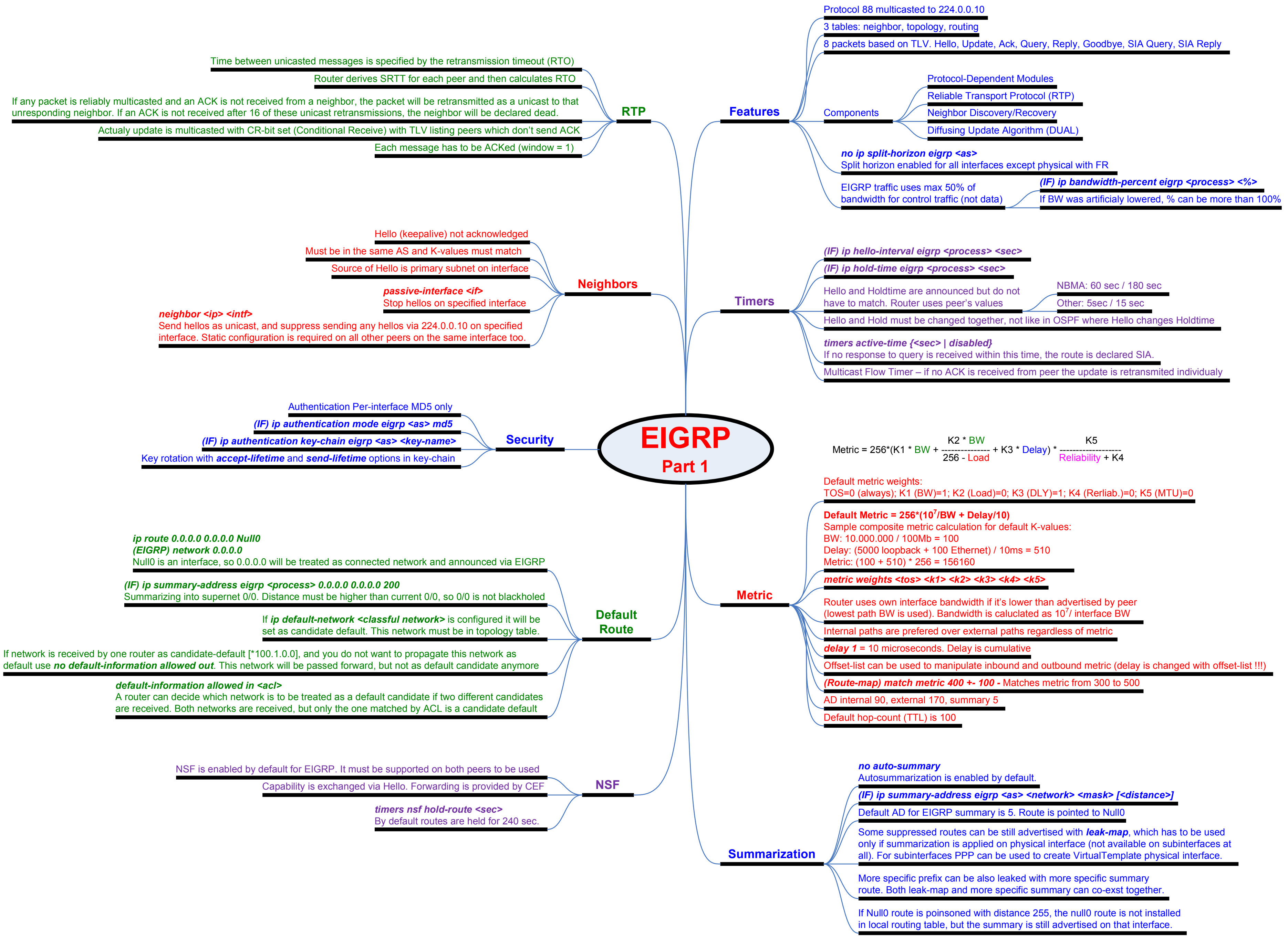
- Update 30 sec
  - The specific random variable used by Cisco IOS, RIP_JITTER, subtracts up to 15 percent (4.5 seconds) from the update time. Therefore, updates from Cisco routers vary between 25.5 and 30 seconds
- Invalid 180 sec
  - Route becomes invalid if no updates for it are heard. Route is marked inaccessible and advertised as unreachable but router still uses it to forward packets.
- Holddown 180 sec
  - If route's metric changes, do not accept new sources of updates until this timer expires. This timer is introduced by CISCO, it is not in RFC.
- Flush (garbage) 240 sec
  - Route is removed if timer expires. Starts with invalid timer
- ***timers basic <update> <invalid> <hold> <flush> <sleep ms>***
  sleep – delays regular periodic update after receiveing a triggered update
- ***flash-update threshold <sec>***
  if this amount of time is left before a full update, triggered update is suppressed
- ***output-delay <sec>***
  if multiple packets are to be sent, wait this time between packets

## Default route

- ***default-information-originate [route-map <name>]***
  Causes injection of 0/0 even if 0/0 does not exist in routing table. Route map can be used to generate a default conditionaly or to set interface out which default can be advertised. It gets metric of 1
- ***ip default-network <major-network>***
  Advertises 0/0 as a default network. The network must be a major network which is localy connected. Ex. For network 100.100.100.0/24 connected to Serial0/0, default-network must be defined as 100.0.0.0
- ***ip route 0.0.0.0 0.0.0.0 null0***
  Default can be injected either with ***redistribute static*** or ***network 0.0.0.0***.
  Neighbor routers set advertising router as a Gateway of last resort
- Default is also automaticaly sent to peers if it's redistributed from other protocols.

## Filtering

- Route is always added to database, but filtered when populating into route table, except routes with infinity metric, which are not even added to database
- ***distribute-list <acl> {in | out} [<if>]***
- ***distribute-list gateway <prefix> {in | out} [<if>]***
  Filter updates from specific sources only. Prefix list must be used to define source list, not ACL.
- ***distribute-list prefix <list> [gateway <prefix>] {in | out} [<if>]***
  Filter specific prefixes from updates from specific sources only. Prefix list must be used in both parts, not ACL.
- ***offset-list <acl> {in | out} <offset> [<if>]***
  Add artificial metric to received or sent updates. If ACL is 0 (zero) then no ACL is used. Can be used to filter updates by adding infinite offset 16. Route is not even added to database, it is dropped. Offset is added to all advertised routes, regardless if they are redistributed or originated by RIP
- ***passive interface <if>***
  disable sending updates, but still receives updates. To filter inbound updates distribute-list must be used



# EIGRP

## Part 1

## RTP

- Time between unicasted messages is specified by the retransmission timeout (RTO)
- Router derives SRTT for each peer and then calculates RTO
- If any packet is reliably multicasted and an ACK is not received from a neighbor, the packet will be retransmitted as a unicast to that unresponding neighbor. If an ACK is not received after 16 of these unicast retransmissions, the neighbor will be declared dead.
- Actualy update is multicasted with CR-bit set (Conditional Receive) with TLV listing peers which don't send ACK
- Each message has to be ACKed (window = 1)

## Neighbors

- Hello (keepalive) not acknowledged
- Must be in the same AS and K-values must match
- Source of Hello is primary subnet on interface
- ***passive-interface <if>***  
  Stop hellos on specified interface
- ***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.

## Security

- Authentication Per-interface MD5 only
- ***(IF) ip authentication mode eigrp <as> md5***
- ***(IF) ip authentication key-chain eigrp <as> <key-name>***
- Key rotation with ***accept-lifetime*** and ***send-lifetime*** options in key-chain

## Default Route

- ***ip route 0.0.0.0 0.0.0.0 Null0***  
  ***(EIGRP) network 0.0.0.0***  
  Null0 is an interface, so 0.0.0.0 will be treated as connected network and announced via EIGRP
- ***(IF) ip summary-address eigrp <process> 0.0.0.0 0.0.0.0 200***  
  Summarizing into supernet 0/0. Distance must be higher than current 0/0, so 0/0 is not blackholed
- If ***ip default-network <classful network>*** is configured it will be set as candidate default. This network must be in topology table.
- 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-information allowed in <acl>***  
  A router can decide which network is to be treated as a default candidate if two different candidates are received. Both networks are received, but only the one matched by ACL is a candidate default

## NSF

- NSF is enabled by default for EIGRP. It must be supported on both peers to be used
- Capability is exchanged via Hello. Forwarding is provided by CEF
- ***timers nsf hold-route <sec>***  
  By default routes are held for 240 sec.

## Features

- Protocol 88 multicasted to 224.0.0.10
- 3 tables: neighbor, topology, routing
- 8 packets based on TLV. Hello, Update, Ack, Query, Reply, Goodbye, SIA Query, SIA Reply
- Components
  - Protocol-Dependent Modules
  - Reliable Transport Protocol (RTP)
  - Neighbor Discovery/Recovery
  - Diffusing Update Algorithm (DUAL)
- ***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)
  - ***(IF) ip bandwidth-percent eigrp <process> <%>***
  - If BW was artificialy lowered, % can be more than 100%

## Timers

- ***(IF) ip hello-interval eigrp <process> <sec>***
- ***(IF) ip hold-time eigrp <process> <sec>***
- Hello and Holdtime are announced but do not have to match. Router uses peer's values
  - NBMA: 60 sec / 180 sec
  - Other: 5sec / 15 sec
- Hello and Hold must be changed together, not like in OSPF where Hello changes Holdtime
- ***timers active-time {<sec> | disabled}***  
  If 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 individualy

## Metric

$$\text{Metric} = 256*(K1 * BW + \frac{K2 * BW}{256 - Load} + K3 * Delay) * \frac{K5}{Reliability + K4}$$

- Default metric weights:  
  TOS=0 (always); K1 (BW)=1; K2 (Load)=0; K3 (DLY)=1; K4 (Rerliab.)=0; K5 (MTU)=0
- **Default Metric = 256*($10^7$/BW + 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
- ***metric weights <tos> <k1> <k2> <k3> <k4> <k5>***
- Router uses own interface bandwidth if it's lower than advertised by peer (lowest path BW is used). Bandwidth is caluclated as $10^7$/ interface BW
- Internal paths are prefered over external paths regardless of metric
- ***delay 1*** = 10 microseconds. Delay is cumulative
- Offset-list can be used to manipulate inbound and outbound metric (delay is changed with offset-list !!!)
- ***(Route-map) match metric 400 +- 100 -*** Matches metric from 300 to 500
- AD internal 90, external 170, summary 5
- Default hop-count (TTL) is 100

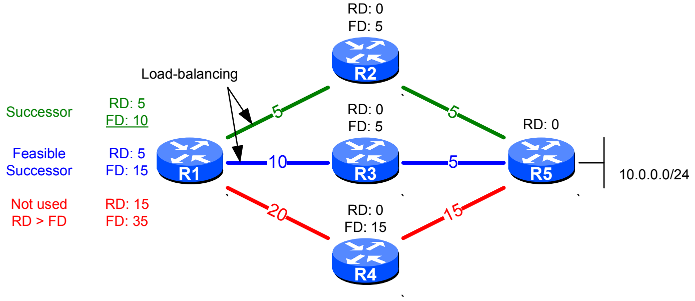
## Summarization

- ***no auto-summary***  
  Autosummarization 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
- 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-exst together.
- If Null0 route is poinsoned with distance 255, the null0 route is not installed in local routing table, but the summary is still advertised on that interface.

# EIGRP
## Part 2

### Load balancing

- ***maximum-paths <1..16>***
  By default EIGRP will loRD balance across 4 eual paths
- ***traffic-share min*** – send traffic over lowest-cost path only
- ***traffic-share min across-interfaces***
  If more paths exist than allowed choose the ones over different physical interfaces
- ***traffic-share balanced*** – less packets to lower-bandwidth paths (default)
- ***variance <multiplier>***
  Multiplier is multiplied by FD (divide the worst route by the best route). Any metric which is lower than this value and meets FS condition is also considered as valid loRD-balanced path.
- ***variance 2***
  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



### Redistribution and filtering

- IP EIGRP automatically redistributes IGRP routes if the IGRP process is in the same autonomous system.
- No default metric, must be manual set when redistributing into EIGRP
- Metric is derived automaticaly for routes redistributed from connected, static or other EIGRP processes
- ***default-metric <bw> <delay> <reliability> <loRD> <mtu>***
- ***redistribute <protocol> metric <bw> <delay> <reliability> <loRD> <mtu>***
- ***distribute-list <acl> {in | out} [<if>]***
- ***distribute-list prefix <name> {in | out} [<if>]***
- Tags can be aded to routes to manipulate route entries and mutual redistribution
- ***metric maximum-hop 1***
  You can filter prefixes to be announced only to nearest peer
- Distance
  - ***distance eigrp <internal> <external>***
    Distance set for all internal and external prefixes
  - ***distance <distance> <source IP> <source mask> [<acl>]***
    Distance set for specific prefixes originated by specific source (works ONLY for internal routes, external are not matched at all)

### Topology (DUAL)

- RD – reported distance (by peer)
- 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)
- Metrics for each route shown as: (Feasible distance / Reported distance)
- ***show ip eigrp topology all-links*** (show non-FS)
- If some route fails
  - **1**. If FS exists, the one with lowest metric is installed and an update is sent to other peers. The FD from the Feasible Successor does not overwrite FD for the prefix itself (FD stays unchanged) unles active query is performed
  - **2**. If no FS exists, router performs active query for prefix
    - Query
      - A query origin flag (O) is set to 1 – router originated query
      - All queries and replies must be ACKed (RTP)
      - When active query is initiated existing FD/RD is set to Infinity, so every new source will be better.
      - For each neighbor to which a query is sent, the router will set a reply status flag (r) to keep track of all outstanding queries
    - **a)** Router multicasts **query** to other peers
    - **b)** Each peer unicasts reply if they have or not, loop-free route to that prefix
    - **c)** Router updates own tolopogy table only if all neighbors replied
    - **d)** If peer doesn't have unchanged FD route of its FS does not exist, it witholds reply and performes own active query to all peers, except the one from which initial query was received. A query origin flag (O) is set to 0 – router received query and stared own query
    - **e)** If router stays too long in active query the route becomes **SIA**
      - ***timers active-time {<time> | disabled}***
        If active Timer (3min) expires All peers which did not reply to query are reset
      - The SIA-retransmit timer is set to one-half the value of the Active timer: 90 seconds
      - The routers will send up to three SIA-queries as long as SIA-replies are received, before resetting a neighbor.

Query scoping is used to avoid SIA and to minimize convergence
- Stub router
- **Route summarization** – if peer does not have queried prefix but it has summarized route it instantly replies negatively without doing own query

### Stub router

- Stub by default announces connected and summary. Connected means covered by network statement or redistributed as connected. Redistributed routes cover only those not covered by network statement.
- ***eigrp stub {connected summary static redistributed receive-only} [leak <route-map>]***
- Routers do not query stub routers at all. Stub is announced in Hello
- Stub routers should not be used as transit
- Leak-map can be used to RDvertise **ANY** RDditional routes (even those learned from other peers, regardless of stub route types to be RDvertised), but querying is still suppressed, as it is a stub.
- Leaked routes can be limited per-neighbor by specyfing interface
  ***route-map LEAK permit 10***
  ***match ip address <acl>***
  ***match interface <if>*** - outgoing interface toward neighbor

# OSPFv2

## Stub Areas

- **Stubby area**
  *area <id> stub*
  Suppress LSA5. generates LSA3 default with cost 1
  - **Totaly stubby**
    *area <id> stub no-summary*
    Configured only on ABR. Suppress LSA3 (except a default)
  - All stub routers set E-bit=0 flag in Hello. Adjacencies will not be set with router not configured as a stub
- **Not-so-stubby (NSSA)**
  *area <id> nssa*
  Suppress LSA5. Default is not generated automaticaly
  - **Totaly Not-so-stubby**
    *area <id> nssa no-summary*
    Configured only on ABR. Suppress LSA3. except LSA3 default which is generated automaticaly with cost 1.
  - Allows external LSA7 translated to LSA5 by ABR.

## Default

- ***area <id> default-cost <cost>***
  Set cost for a default route automaticaly generated by an ABR. Useful if many ABRs exist. By default cost of default is 1
- ***area <id> nssa no-summary default-information-originate***
  Default will be originated as N2 with cost 1. Overrides no-summary LSA3 generation
- ***area <id> nssa default-information-originate***
  If ***no-summary*** from NSSA is removed, default can be originated as N2
- If regular router originates default it becomes ASBR. If ABR originates default it is not an ASBR
- OSPF does not support summary-address 0.0.0.0 to generate a default

## Modes

- **P-to-P**
  - NO DR and BDR election. Hello sent as **multicast** (10 / 40)
- **Non-broadcast**
  - DR and BDR election. Hello sent as **unicast** (30 / 120)
  - ***interface serial0/0.1 multipoint*** – NBMA, NOT p-t-multipoint!!!
  - ***neighbor <ip> [priority <id>] [poll-interval <sec>]***
    Static neighbor configuration is required (usualy only on Hub)
  - DR passes routes along but does not change any lookup attributes (next-hop), so static L2/L3 mapping is required on FR but without broadcast keyword
  - Priority for spokes should be 0 so spokes will not become DR/BDR when Hub flaps
- **broadcast**
  - NH not changed on Hub-Spoke FR, so L2/L3 mapping is required for spokes to communicate (with broadcast keyword)
  - ***ip ospf network broadcast***
  - DR and BDR election. Hello sent as **multicast** (10 / 40)
- **P-t-Mpoint**
  - NO DR and BDR election. Hello sent as **multicast** (30 / 120). PollInterval is 120 sec.
  - Networks are treated as a collection of point-to-point links
  - Hub router changes FA to itself when passing routes between spokes
  - ***ip ospf network point-to-multipoint*** - on each router, as timers are changed
  - If static L2/L3 mapping is used broadcast keyword must be used
  - The segment is seen as collection of /32 endpoints (regardless of netmask), not a transit subnet
  - **Non-broadcast**
    - Used for unequal spokes. Cost for neighbor can be assigned only in this type
    - Hellos unicasted. Broadcast keyword is not required for static L2/L3 mapping

## Cost

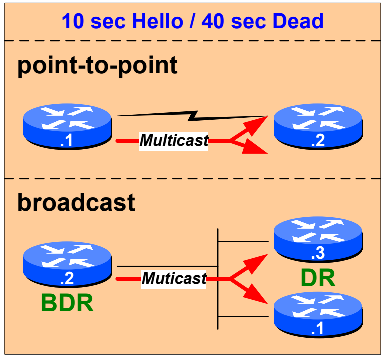
- Default autocost reference: 100.000.000/BW bps
- *auto-cost reference-bandwidth <bw in Mbps>*
- **Refrerence = Cost * BW (Mbps) – default 100**
- ***(IF) ip ospf cost <cost>***
- ***neighbor <ip> cost <cost>***
  only for point-to-multipoint and point-to-multipoint non-broadcast type (spokes with different CIRs)

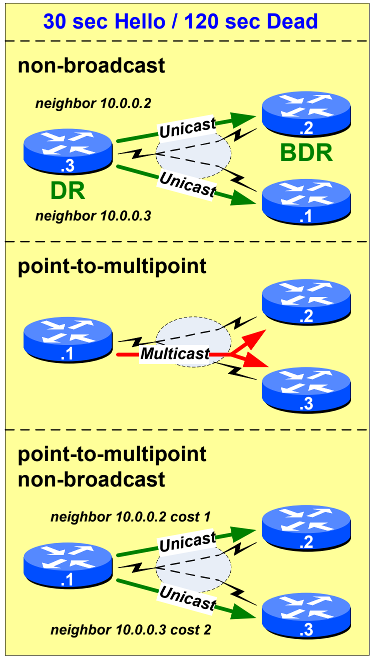
## Features

- IP protocol 89; 224.0.0.5 All OSPF Routers; 224.0.0.6 All DR Routers
- Router-ID can be any dotted-decimal number (0.0.0.1), not necessarily valid IP
- Router ID can be the same with different areas, but not for ASBR
- Route selection: 1. Intra-area; 2. Inter-area; 3. External E1; 4. External E2
- Metric is compared only if routes are of the same type

## Timers

- Hello: 10 sec LAN, 30 sec NBMA; Dead: 4x Hello (40 sec LAN, 120 sec NBMA) – counts down
- LSARefresh: 30 min - Each router originating LSA re-floods id with incremented Seq every 30 min (Link State Refresh interval)
- LSA Maxage: 60 min - Each router expects LSA to be refreshed within 60 min
- 1sec Dead with 250ms Hello (Fast Hello Feature):
  ***(IF) ip ospf dead-interval minimal hello mutiplier 4***
- ***(IF) ip ospf retransmit-interval <sec>*** - time between LSUs (if not ACKed) default 5 sec
- ***(IF) ip ospf hello-interval <sec>*** - Hold will be automaticaly set to 4x Hello
- ***(IF) ip ospf transmit-delay <sec>***
  age is incremented by a InfTransDelay ( default 1sec) on transited routers. It is also incremented as it resides in the database.
- Poll interval: on NBMA Hello to neighbor marked down – 60 sec
- Pacing
  - ***timers pacing retransmission <msec>***
    Time at which LSA in retransmission queue are paced – 66ms
  - ***timers pacing flood <msec>***
    Time in msec between consecutive LSUs when flooding LSA – 33 msec
  - ***timers pacing lsa-group <sec>***
    By delaying the refresh, more LSAs can be grouped together (default 240 sec)
- ***(IF) ip ospf flood-reduction***
  Stop LSA flooding every 30 min by setting DoNotAge flag, removing requirement for periodic refresh on point-to-point links





| *ip ospf network* | DR<br>BDR | Hello<br>Int | static<br>nghbr | Hello<br>Type |
|---|---|---|---|---|
| ***broadcast*** (Cisco) | Y | 10 | N | Mcast |
| ***point-to-point*** (Cisco) | N | 10 | N | Mcast |
| ***nonbroadcast*** (Phy FR) (RFC) | Y | 30 | Y | Unicast |
| ***point-to-multipoint*** (RFC) | N | 30 | N | Mcast |
| ***point-to-multipoint nonbr*** (Cisco) | N | 30 | Y | Unicast |

# OSPF Filtering

## Summarization

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

***summary-address <prefix> <mask> [no-advertise] [tag <tag>]***
Extenral routes can be summarized only on ASBR which redistributed those routes. Cost is taken from smallest cost of component routes

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

***no discard-route {internal | extenral}***
Since 12.1 summary will automaticaly create null0 route to prevent loops. It can be disabled

***area <id> nssa translate type7 suppress-fa***
If summarization is used FA is lost in NSSA. ABR sets FA to 0.0.0.0, what means that other routers will use ABR as FA

Additional summary can be created for that more specific route (multiple summaries)

## DB overload protection

***redistribute max-prefix <max routes> <% warning> [warning-only]***
Only external routes are counted. After warning level is reached, routes are still accepted, but message is re-sent to syslog

***max-lsa <max routes> <% warning> [warning-only] [ignore-time <min>] [ignore-count <#>] [reset-time <min>]***
Only internal, non-self-originated routes are counted. When the ***warning-only*** 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.

## Stub router

The router will not be used as transit, unless it is the only path

***max-metric router-lsa on-startup {<announce-time> | wait-for-bgp}***
Advertises max metric for all routes, which are not originated by that router

Local routes are advertised with normal metric

## Virtual-Link

***area <transit-area> virtual-link <RID of ABR connecting to area 0>***
Configured on ABRs

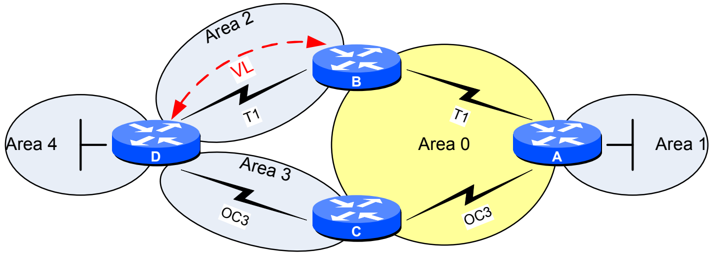
VL can stay active after authentication is applied as it is an on-demand circuit (hellos suppressed)

VL cannot be used over Stub area, but GRE tunnel can

VL is an interface in area 0 (must be authenicated if area 0 is authenticated)

VL has no IP address, so it does not carry data traffic, only control-plane

The best path from D to A is through OC3 links via C. Normaly, D would sent traffic through area 0 via B (VL is in area 0). However, ***capability transit*** (enabled by default) causes the best path to be choosen via C. If this feature is disabled traffic always goes through area 2



## Prefix suppression

When OSPF is enabled on the interface, it always advertises directly connected subnet.
To stop advertisement the link can be set as unnumbered or preffix can be suppressed

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

***(IF) ip ospf prefix-suppression [disable]***
Suppress all prefixes on interface (loopbacks and passive too). Takes precedence over router-mode command.
Disable keyword makes OSPF advertise the interface ip prefix, regardless of router mode configuration

## Redistribution and route origin

If „subnets" keyword is omited, router redistributes classful subnets, not classful versions of subnets (1.0.0.0/8 will be advertised, 131.0.0.0/24 will not)

OSPF default metric (E2) of redistributed IGP routes=20 (subnets) and 1 for BGP

```
router ospf <process>
 network <net> <wildcard> area <id>
```
Secondary subnets on interface <u>covered by the network</u> command are advertised as Stub (non-transit, no LSA2) only if primary is also advertised. If an interface is unnumbered, and network matches primary intf, OSPF is enabled also on unnumbered (hellos are sent)

```
interface fastethernet0/0
 ip ospf <process> area <id>
```
Any and all interface secondary subnets are advertised unless:
```
 ip ospf <process> area <id> secondaries none
```

## Filtering

### *distribute-list*

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 only exception to „in" is when prefix being filtered is comming from area 0, then prefix will be filtered from routing table AND a database

„Out" works only on any ASBR or also on ABR if area is NSSA. Used to filter ONLY LSA5 and LSA7 from DATABASE. Local router still has the prefix in routing table, but it is not announced to peers.

If interface is included it is treated as outgoing interface for NH of matched route, and only such route will be considered

If route-map is used, route can be matched with „***match ip route-source <acl>***" matching RID, not NH

### LSA3 on ABR

***area <id> filter-list prefix <name> {in | out}***
in – into area, out – outside area (into area0)

Configured on ABR at the point where LSA3 would be created. Filters ONLY LSA3

***not-advertise*** in area-range – No LSA3 is propagated. The effect is the same as filter-list. Only LSA1 is filtered

***not-advertise*** in summary – Only LSA5/7 is filtered from database

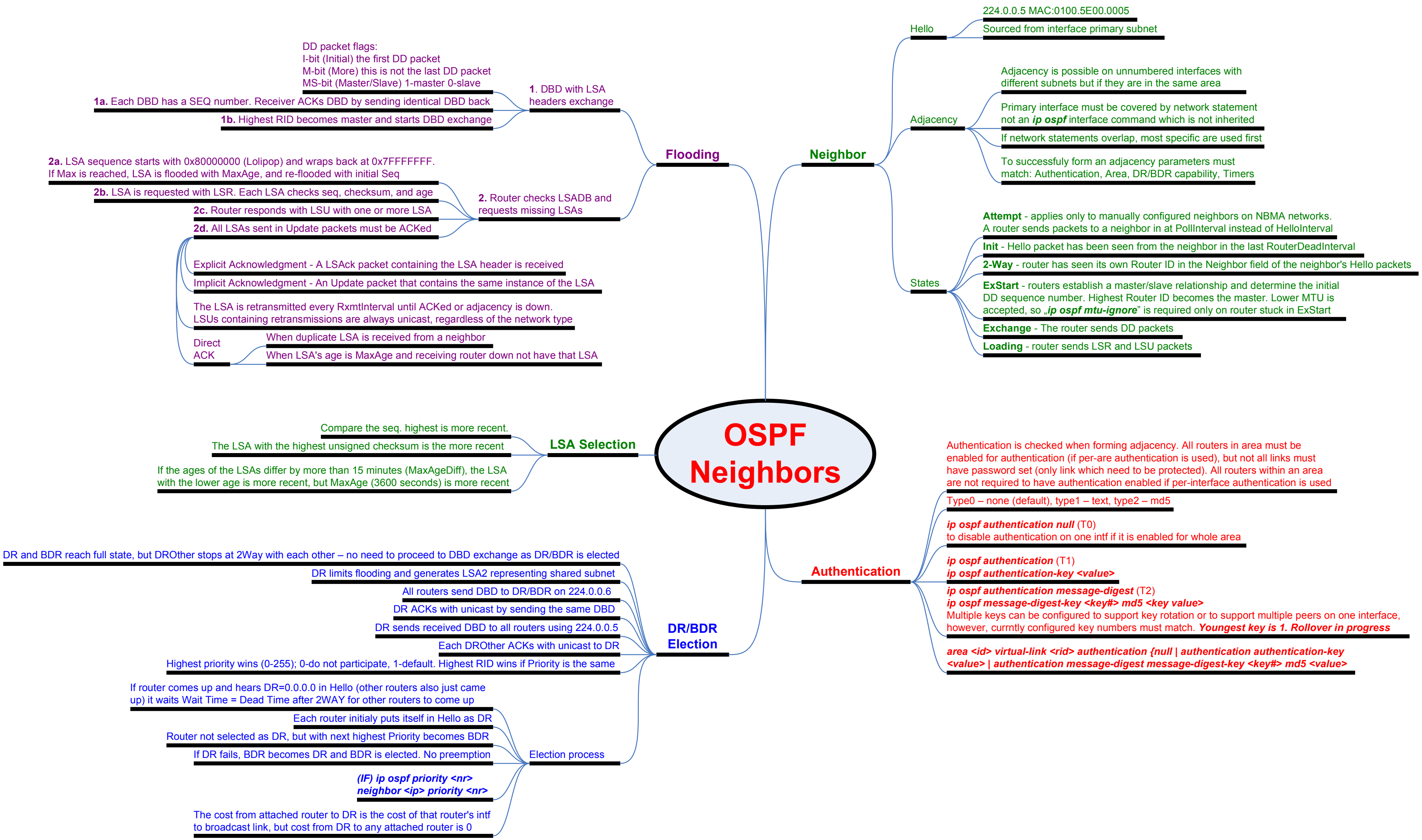
### Database filtering

All outgoing LSAs are filtered.

***(IF) ip ospf database-filter out***
On multipoint interface, all neighbors are filtered

***neighbor <ip> database-filter-all out***
Only on p-2-mpoint interface, per neighbor

***area <id> nssa no-redistribution***
Used if the same router is ABR and ASBR at the same time, and there is no need to redistribute routes into nssa (especialy if no-summary is used). Routes are then redistributed only to area 0 as LSA5, but not into NSSA area as LSA7. Useful if ABR is the only exit point from NSSA area.



# OSPF Neighbors

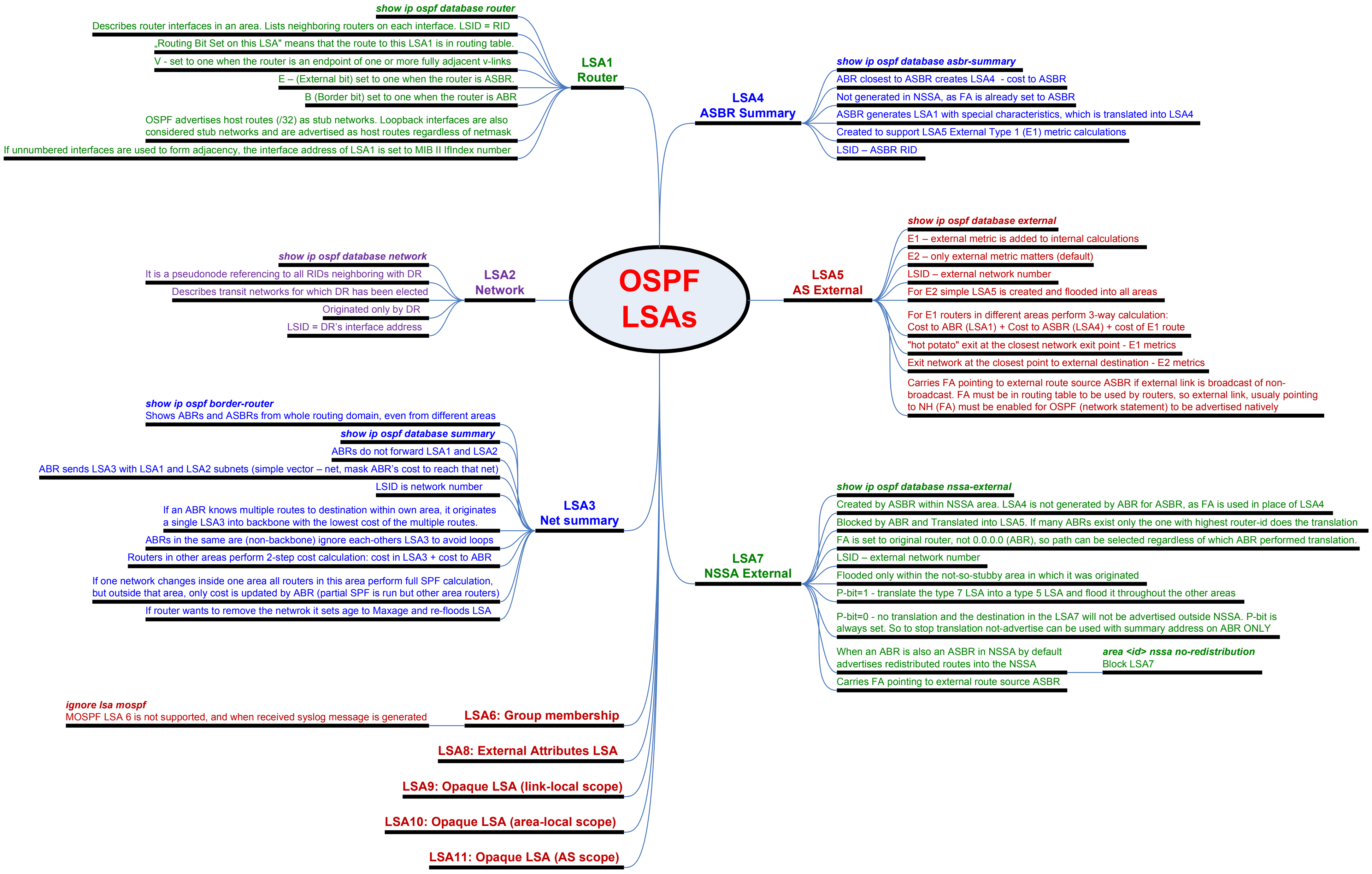
- **Flooding**
  - **1**. DBD with LSA headers exchange
    - DD packet flags:
      I-bit (Initial) the first DD packet
      M-bit (More) this is not the last DD packet
      MS-bit (Master/Slave) 1-master 0-slave
    - **1a.** Each DBD has a SEQ number. Receiver ACKs DBD by sending identical DBD back
    - **1b.** Highest RID becomes master and starts DBD exchange
  - **2.** Router checks LSADB and requests missing LSAs
    - **2a.** LSA sequence starts with 0x80000000 (Lolipop) and wraps back at 0x7FFFFFFF. If Max is reached, LSA is flooded with MaxAge, and re-flooded with initial Seq
    - **2b.** LSA is requested with LSR. Each LSA checks seq, checksum, and age
    - **2c.** Router responds with LSU with one or more LSA
    - **2d.** All LSAs sent in Update packets must be ACKed
      - 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 RxmtInterval 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 down not have that LSA
- **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
    - If network statements overlap, most specific are used first
    - To successfully form an adjacency parameters must match: Authentication, Area, DR/BDR capability, Timers
  - States
    - **Attempt** - applies only to manually configured neighbors on NBMA networks. A router sends packets to a neighbor in at PollInterval instead of HelloInterval
    - **Init** - Hello packet has been seen from the neighbor in the last RouterDeadInterval
    - **2-Way** - router has seen its own Router ID in the Neighbor field of the neighbor's Hello packets
    - **ExStart** - routers establish a master/slave relationship and determine the initial DD sequence number. Highest Router ID becomes the master. Lower MTU is accepted, so „***ip ospf mtu-ignore***" is required only on router stuck in ExStart
    - **Exchange** - The router sends DD packets
    - **Loading** - router sends LSR and LSU packets
- **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 (3600 seconds) is more recent
- **Authentication**
  - Authentication is checked when forming adjacency. All routers in area must be enabled for authentication (if per-are 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
  - Type0 – none (default), type1 – text, type2 – md5
  - ***ip ospf authentication null*** (T0)
    to disable authentication on one intf if it is enabled for whole area
  - ***ip ospf authentication*** (T1)
    ***ip ospf authentication-key <value>***
  - ***ip ospf authentication message-digest*** (T2)
    ***ip ospf message-digest-key <key#> md5 <key value>***
    Multiple keys can be configured to support key rotation or to support multiple peers on one interface, however, currntly configured key numbers must match. ***Youngest key is 1. Rollover in progress***
  - ***area <id> virtual-link <rid> authentication {null | authentication authentication-key <value> | authentication message-digest message-digest-key <key#> md5 <value>***
- **DR/BDR Election**
  - DR and BDR reach full state, but DROther stops at 2Way with each other – no need to proceed to DBD exchange as DR/BDR is elected
  - DR limits flooding and generates LSA2 representing shared subnet
  - All routers send DBD to DR/BDR on 224.0.0.6
  - DR ACKs with unicast by sending the same DBD
  - DR sends received DBD to all routers using 224.0.0.5
  - Each DROther ACKs with unicast to DR
  - Highest priority wins (0-255); 0-do not participate, 1-default. Highest RID wins if Priority is the same
  - Election process
    - If router comes up and hears DR=0.0.0.0 in Hello (other routers also just came up) it waits Wait Time = Dead Time after 2WAY for other routers to come up
    - Each router initialy puts itself in Hello as DR
    - Router not selected as DR, but with next highest Priority becomes BDR
    - If DR fails, BDR becomes DR and BDR is elected. No preemption
    - ***(IF) ip ospf priority <nr>***
      ***neighbor <ip> priority <nr>***
    - The cost from attached router to DR is the cost of that router's intf to broadcast link, but cost from DR to any attached router is 0

| | |
|---|---|
| O | intra-area |
| O IA | inter-area (LSA3) |
| O E1 | external type 1 (LSA5) |
| O E2 | external type 2 (LSA5) |
| O N1 | NSSA external type 1 (LSA7) |
| O N2 | NSSA external type 2 (LSA7) |

| Area | 1&2 | 3 | 4 | 5 | 7 |
|---|---|---|---|---|---|
| Area 0 | Yes | Yes | Yes | Yes | No |
| Regular | Yes | Yes | Yes | Yes | No |
| Stub | Yes | Yes | No | No | No |
| Totally | Yes | No* | No | No | No |
| NSSA | Yes | Yes | Yes | No | Yes |

*Except LSA3 default route (IA)

| Area | Stop LSA5 | Stop LSA3 | Create LSA7 |
|---|---|---|---|
| stub | Y | N | N |
| totaly stub | Y | Y | N |
| nssa | Y | N | Y |
| totaly nssa | Y | Y | Y |



# OSPF LSAs

## LSA1 Router

*show ip ospf database router*

Describes router interfaces in an area. Lists neighboring routers on each interface. LSID = RID

„Routing Bit Set on this LSA" means that the route to this LSA1 is in routing table.

V - set to one when the router is an endpoint of one or more fully adjacent v-links

E – (External bit) set to one when the router is ASBR.

B (Border bit) set to one when the router is ABR

OSPF advertises host routes (/32) as stub networks. Loopback interfaces are also considered stub networks and are advertised as host routes regardless of netmask

If unnumbered interfaces are used to form adjacency, the interface address of LSA1 is set to MIB II IfIndex number

## LSA2 Network

*show ip ospf database network*

It is a pseudonode referencing to all RIDs neighboring with DR

Describes transit networks for which DR has been elected

Originated only by DR

LSID = DR's interface address

## LSA3 Net summary

*show ip ospf border-router*

Shows ABRs and ASBRs from whole routing domain, even from different areas

*show ip ospf database summary*

ABRs do not forward LSA1 and LSA2

ABR sends LSA3 with LSA1 and LSA2 subnets (simple vector – net, mask ABR's cost to reach that net)

LSID is network number

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.

ABRs in the same are (non-backbone) ignore each-others LSA3 to avoid loops

Routers in other areas perform 2-step cost calculation: cost in LSA3 + cost to ABR

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)

If router wants to remove the netwrok it sets age to Maxage and re-floods LSA

## LSA4 ASBR Summary

*show ip ospf database asbr-summary*

ABR closest to ASBR creates LSA4 - cost to ASBR

Not generated in NSSA, as FA is already set to ASBR

ASBR generates LSA1 with special characteristics, which is translated into LSA4

Created to support LSA5 External Type 1 (E1) metric calculations

LSID – ASBR RID

## LSA5 AS External

*show ip ospf database external*

E1 – external metric is added to internal calculations

E2 – only external metric matters (default)

LSID – external network number

For E2 simple LSA5 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

"hot potato" exit at the closest network exit point - E1 metrics

Exit network at the closest point to external destination - E2 metrics

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, usualy pointing to NH (FA) must be enabled for OSPF (network statement) to be advertised natively

## LSA7 NSSA External

*show ip ospf database nssa-external*

Created by ASBR within NSSA area. LSA4 is not generated by ABR for ASBR, as FA is used in place of LSA4

Blocked by ABR and Translated into LSA5. If many ABRs exist only the one with highest router-id does the translation

FA is set to original router, not 0.0.0.0 (ABR), so path can be selected regardless of which ABR performed translation.

LSID – external network number

Flooded only within the not-so-stubby area in which it was originated

P-bit=1 - translate the type 7 LSA into a type 5 LSA and flood it throughout the other areas

P-bit=0 - no translation and the destination in the LSA7 will not be advertised outside NSSA. P-bit is always set. So to stop translation not-advertise can be used with summary address on ABR ONLY

When an ABR is also an ASBR in NSSA by default advertises redistributed routes into the NSSA

*area &lt;id&gt; nssa no-redistribution*

Block LSA7

Carries FA pointing to external route source ASBR

## LSA6: Group membership

*ignore lsa mospf*

MOSPF LSA 6 is not supported, and when received syslog message is generated

## LSA8: External Attributes LSA

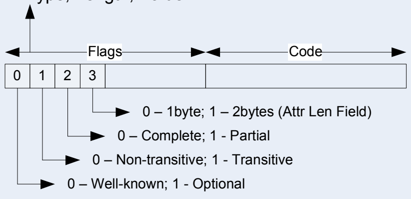
## LSA9: Opaque LSA (link-local scope)

## LSA10: Opaque LSA (area-local scope)

## LSA11: Opaque LSA (AS scope)

## Path arrtibutes

<Type, Length, Value>


Flags
Code
0
1
2
3
0 – 1byte; 1 – 2bytes (Attr Len Field)
0 – Complete; 1 - Partial
0 – Non-transitive; 1 - Transitive
0 – Well-known; 1 - Optional

| | | | |
|---|---|---|---|
| 1 | Origin | | WK M |
| 2 | AS_Path | | WK M |
| 3 | Next_Hop | WK M | |
| 4 | MED | | O NT |
| 5 | Local_Pref | | WK D |
| 6 | Atomin_Aggregate | WK D | |
| 7 | Aggregator | | O T |
| 8 | Community | | O T |
| 9 | Originator_ID | | O NT |
| 10 | Cluster_List | | |
| 12 | Advertiser | | |
| 13 | RCID_Path/Cluster_Id | | |
| 14 | MP-reachable NLRI | O NT | |
| 15 | MP-unreachable NLRI | | O NT |
| 16 | Extended Communities | | |

## RegExp

| | |
|---|---|
| **.** | **Single character** |
| ***** | **Zero or more** |
| **+** | **One or more** |
| **?** | **Zero or one** |
| **[]** | **Range** |
| **[^]** | **Negate range** |
| **^** | **Begining of input** |
| **$** | **End of input** |
| **_** | **, { } ( ) ^ $, space** |
| **\** | **Escape special character** |
| **\\1** | **Repeat a match in ()** |
| **\|** | **Logical OR** |

## Decision Process

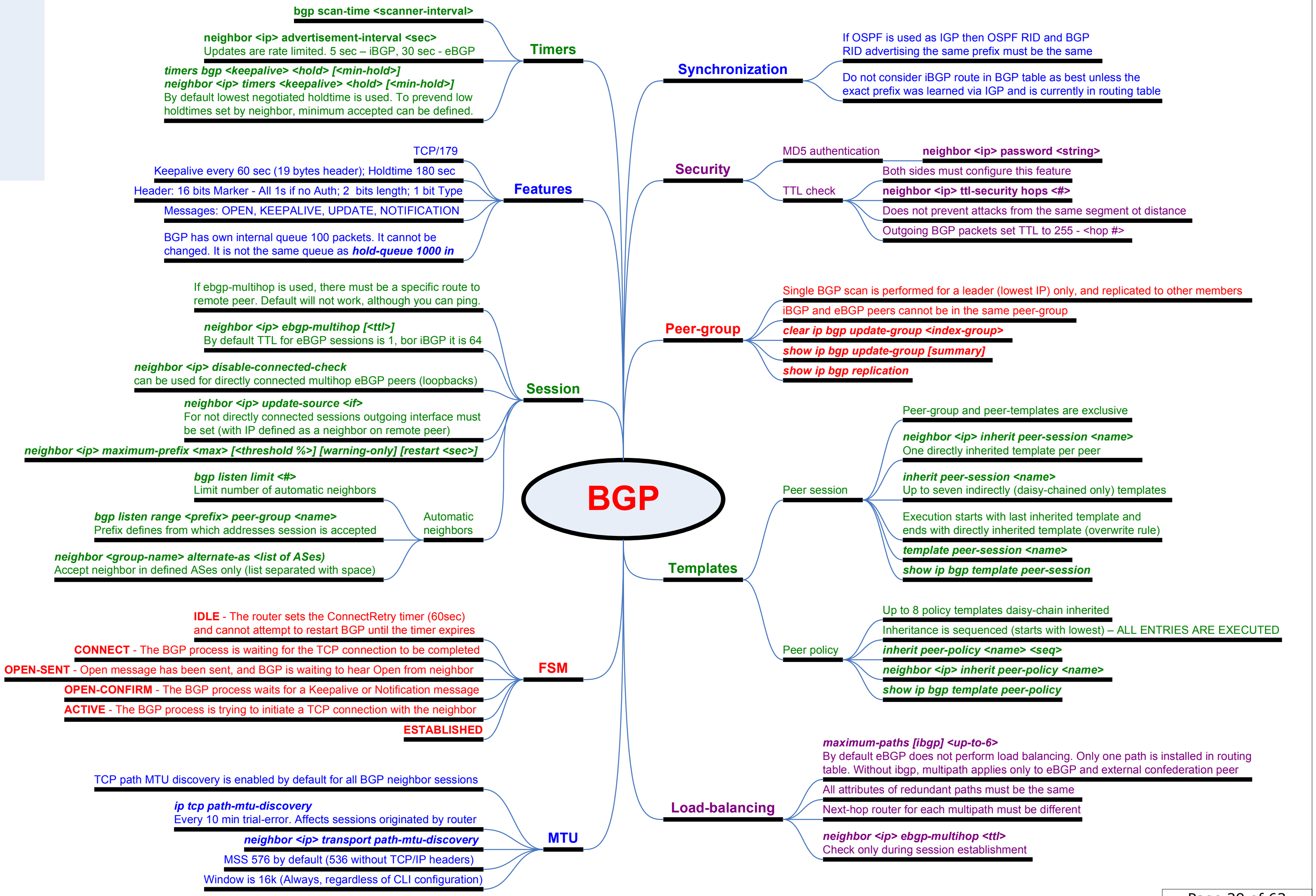
**1. Largest Weight** (localy originated paths: 32768, other 0)
**2. Largest Local-Preefernce** („*bgp default local-preference*”) default 100
**3. Prefer local paths** (decreasing preference: *default-originate* in neighbor, *default-information-originate* in global, *network, redistribute, aggrgegate*)
**4. Shortest AS_PATH** („*bgp bestpath as-path ignore”* bypasses this step; AS_SET counts as 1; AS_CONFED_SEQUENCE and AS_CONFED_SET are not counted)
**5. Lowest origin code** (0-IGP, 1-EGP, 2-Incomplete)
**6. Lowest MED** (*bgp always-compare-med*; *bgp bestpath med-confed*; *bgp bestpath med missing-as-worst*; *bgp deterministic-med*) default 0
**7. eBGP prefered over iBGP** (Confed. paths are treated as internal paths)
**8. Closest IGP neighbor** (best cost)
**9. Determine if multiple paths require installation** (multipath)
**10. If paths are external choose the oldest one** (flap prevention). Skipped if „*bgp bestpath compare-routerid*”)
**11. Lowest Router-ID**
**12. Minimum Cluster-List length** (RR environment)
**13. Lowest neighbor address**

## Filter Sequence

IN:
1. ROUTE-MAP
2. FILTER-LIST
3. PREFIX-LIST, DISTRIBUTE-LIST

OUT:
1. PREFIX-LIST, DISTRIBUTE-LIST
2. FILTER-LIST
3. ROUTE-MAP

IGP:
1. DISTRIBUTE-LIST OUT
2. ROUTE-MAP (REDISTRIBUTION)



**Timers**
- ***bgp scan-time <scanner-interval>***
- ***neighbor <ip> advertisement-interval <sec>*** Updates are rate limited. 5 sec – iBGP, 30 sec - eBGP
- ***timers bgp <keepalive> <hold> [<min-hold>]*** ***neighbor <ip> timers <keepalive> <hold> [<min-hold>]*** By default lowest negotiated holdtime is used. To prevend low holdtimes set by neighbor, minimum accepted can be defined.

**Features**
- TCP/179
- Keepalive every 60 sec (19 bytes header); Holdtime 180 sec
- Header: 16 bits Marker - All 1s if no Auth; 2 bits length; 1 bit Type
- Messages: OPEN, KEEPALIVE, UPDATE, NOTIFICATION
- BGP has own internal queue 100 packets. It cannot be changed. It is not the same queue as ***hold-queue 1000 in***

**Session**
- If ebgp-multihop is used, there must be a specific route to remote peer. Default will not work, although you can ping.
- ***neighbor <ip> ebgp-multihop [<ttl>]*** By default TTL for eBGP sessions is 1, bor iBGP it is 64
- ***neighbor <ip> disable-connected-check*** can be used for directly connected multihop eBGP peers (loopbacks)
- ***neighbor <ip> update-source <if>*** For not directly connected sessions outgoing interface must be set (with IP defined as a neighbor on remote peer)
- ***neighbor <ip> maximum-prefix <max> [<threshold %>] [warning-only] [restart <sec>]***
- Automatic neighbors
  - ***bgp listen limit <#>*** Limit number of automatic neighbors
  - ***bgp listen range <prefix> peer-group <name>*** Prefix defines from which addresses session is accepted
  - ***neighbor <group-name> alternate-as <list of ASes)*** Accept neighbor in defined ASes only (list separated with space)

**FSM**
- **IDLE** - The router sets the ConnectRetry timer (60sec) and cannot attempt to restart BGP until the timer expires
- **CONNECT** - The BGP process is waiting for the TCP connection to be completed
- **OPEN-SENT** - Open message has been sent, and BGP is waiting to hear Open from neighbor
- **OPEN-CONFIRM** - The BGP process waits for a Keepalive or Notification message
- **ACTIVE** - The BGP process is trying to initiate a TCP connection with the neighbor
- **ESTABLISHED**

**MTU**
- TCP path MTU discovery is enabled by default for all BGP neighbor sessions
- ***ip tcp path-mtu-discovery*** Every 10 min trial-error. Affects sessions originated by router
- ***neighbor <ip> transport path-mtu-discovery***
- MSS 576 by default (536 without TCP/IP headers)
- Window is 16k (Always, regardless of CLI configuration)

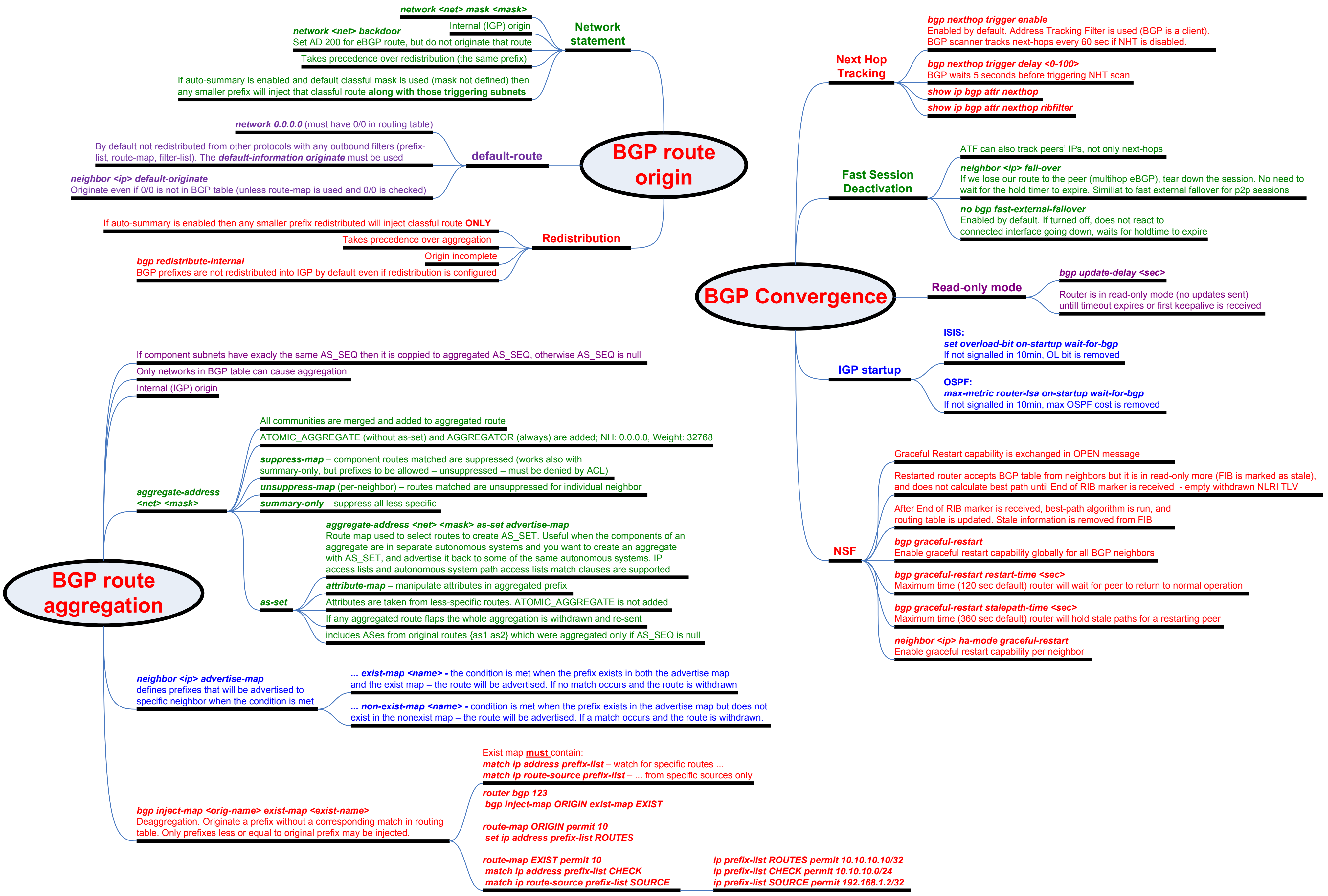
**Synchronization**
- If OSPF is used as IGP then OSPF RID and BGP RID advertising the same prefix must be the same
- Do not consider iBGP route in BGP table as best unless the exact prefix was learned via IGP and is currently in routing table

**Security**
- MD5 authentication
  - ***neighbor <ip> password <string>***
  - Both sides must configure this feature
- TTL check
  - ***neighbor <ip> ttl-security hops <#>***
  - Does not prevent attacks from the same segment ot distance
  - Outgoing BGP packets set TTL to 255 - <hop #>

**Peer-group**
- Single BGP scan is performed for a leader (lowest IP) only, and replicated to other members
- iBGP and eBGP peers cannot be in the same peer-group
- ***clear ip bgp update-group <index-group>***
- ***show ip bgp update-group [summary]***
- ***show ip bgp replication***

**Templates**
- Peer session
  - Peer-group and peer-templates are exclusive
  - ***neighbor <ip> inherit peer-session <name>*** One directly inherited template per peer
  - ***inherit peer-session <name>*** Up to seven indirectly (daisy-chained only) templates
  - Execution starts with last inherited template and ends with directly inherited template (overwrite rule)
  - ***template peer-session <name>***
  - ***show ip bgp template peer-session***
- Peer policy
  - Up to 8 policy templates daisy-chain inherited
  - Inheritance is sequenced (starts with lowest) – ALL ENTRIES ARE EXECUTED
  - ***inherit peer-policy <name> <seq>***
  - ***neighbor <ip> inherit peer-policy <name>***
  - ***show ip bgp template peer-policy***

**Load-balancing**
- ***maximum-paths [ibgp] <up-to-6>*** By default eBGP does not perform load balancing. Only one path is installed in routing table. Without ibgp, multipath applies only to eBGP and external confederation peer
- All attributes of redundant paths must be the same
- Next-hop router for each multipath must be different
- ***neighbor <ip> ebgp-multihop <ttl>*** Check only during session establishment



## BGP route origin

### Network statement

- ***network <net> mask <mask>***
- Internal (IGP) origin
- ***network <net> backdoor***
  Set AD 200 for eBGP route, but do not originate that route
- Takes precedence over redistribution (the same prefix)
- If auto-summary is enabled and default classful mask is used (mask not defined) then any smaller prefix will inject that classful route **along with those triggering subnets**

### default-route

- ***network 0.0.0.0*** (must have 0/0 in routing table)
- By default not redistributed from other protocols with any outbound filters (prefix-list, route-map, filter-list). The ***default-information originate*** must be used
- ***neighbor <ip> default-originate***
  Originate even if 0/0 is not in BGP table (unless route-map is used and 0/0 is checked)

### Redistribution

- If auto-summary is enabled then any smaller prefix redistributed will inject classful route **ONLY**
- Takes precedence over aggregation
- Origin incomplete
- ***bgp redistribute-internal***
  BGP prefixes are not redistributed into IGP by default even if redistribution is configured

## BGP Convergence

### Next Hop Tracking

- ***bgp nexthop trigger enable***
  Enabled by default. Address Tracking Filter is used (BGP is a client).
  BGP scanner tracks next-hops every 60 sec if NHT is disabled.
- ***bgp nexthop trigger delay <0-100>***
  BGP waits 5 seconds before triggering NHT scan
- ***show ip bgp attr nexthop***
- ***show ip bgp attr nexthop ribfilter***

### Fast Session Deactivation

- ATF can also track peers' IPs, not only next-hops
- ***neighbor <ip> fall-over***
  If we lose our route to the peer (multihop eBGP), tear down the session. No need to wait for the hold timer to expire. Similiat to fast external fallover for p2p sessions
- ***no bgp fast-external-fallover***
  Enabled by default. If turned off, does not react to connected interface going down, waits for holdtime to expire

### Read-only mode

- ***bgp update-delay <sec>***
- Router is in read-only mode (no updates sent) untill timeout expires or first keepalive is received

### IGP startup

- **ISIS:**
  ***set overload-bit on-startup wait-for-bgp***
  If not signalled in 10min, OL bit is removed
- **OSPF:**
  ***max-metric router-lsa on-startup wait-for-bgp***
  If not signalled in 10min, max OSPF cost is removed

### NSF

- Graceful Restart capability is exchanged in OPEN message
- Restarted router accepts BGP table from neighbors but it is in read-only more (FIB is marked as stale), and does not calculate best path until End of RIB marker is received - empty withdrawn NLRI TLV
- After End of RIB marker is received, best-path algorithm is run, and routing table is updated. Stale information is removed from FIB
- ***bgp graceful-restart***
  Enable graceful restart capability globally for all BGP neighbors
- ***bgp graceful-restart restart-time <sec>***
  Maximum time (120 sec default) router will wait for peer to return to normal operation
- ***bgp graceful-restart stalepath-time <sec>***
  Maximum time (360 sec default) router will hold stale paths for a restarting peer
- ***neighbor <ip> ha-mode graceful-restart***
  Enable graceful restart capability per neighbor

## BGP route aggregation

- If component subnets have exacly the same AS_SEQ then it is coppied to aggregated AS_SEQ, otherwise AS_SEQ is null
- Only networks in BGP table can cause aggregation
- Internal (IGP) origin
- ***aggregate-address <net> <mask>***
  - All communities are merged and added to aggregated route
  - ATOMIC_AGGREGATE (without as-set) and AGGREGATOR (always) are added; NH: 0.0.0.0, Weight: 32768
  - ***suppress-map*** – component routes matched are suppressed (works also with summary-only, but prefixes to be allowed – unsuppressed – must be denied by ACL)
  - ***unsuppress-map*** (per-neighbor) – routes matched are unsuppressed for individual neighbor
  - ***summary-only*** – suppress all less specific
  - ***as-set***
    - ***aggregate-address <net> <mask> as-set advertise-map***
      Route map used to select routes to create AS_SET. Useful when the components of an aggregate are in separate autonomous systems and you want to create an aggregate with AS_SET, and advertise it back to some of the same autonomous systems. IP access lists and autonomous system path access lists match clauses are supported
    - ***attribute-map*** – manipulate attributes in aggregated prefix
    - Attributes are taken from less-specific routes. ATOMIC_AGGREGATE is not added
    - If any aggregated route flaps the whole aggregate is withdrawn and re-sent
    - includes ASes from original routes {as1 as2} which were aggregated only if AS_SEQ is null
- ***neighbor <ip> advertise-map***
  defines prefixes that will be advertised to specific neighbor when the condition is met
  - ***... exist-map <name> -*** the condition is met when the prefix exists in both the advertise map and the exist map – the route will be advertised. If no match occurs and the route is withdrawn
  - ***... non-exist-map <name> -*** condition is met when the prefix exists in the advertise map but does not exist in the nonexist map – the route will be advertised. If a match occurs and the route is withdrawn.
- ***bgp inject-map <orig-name> exist-map <exist-name>***
  Deaggregation. Originate a prefix without a corresponding match in routing table. Only prefixes less or equal to original prefix may be injected.
  - Exist map **must** contain:
    ***match ip address prefix-list*** – watch for specific routes ...
    ***match ip route-source prefix-list*** – ... from specific sources only
  - 
```
router bgp 123
 bgp inject-map ORIGIN exist-map EXIST

route-map ORIGIN permit 10
 set ip address prefix-list ROUTES

route-map EXIST permit 10
 match ip address prefix-list CHECK
 match ip route-source prefix-list SOURCE
```
```
ip prefix-list ROUTES permit 10.10.10.10/32
ip prefix-list CHECK permit 10.10.10.0/24
ip prefix-list SOURCE permit 192.168.1.2/32
```

# BGP Filtering

## Prefix List

- Autoincrement by 5
- ***ip prefix-list <name> [seq <seq>] permit|deny <prefix> [ge <bits>] [le <bits>]***
- ***neighbor <ip> prefix-list <id> in|out***
- ***distribute-list prefix-list <id> out <routing-process>***
- ***show ip prefix-list [detail | summary]***
- ***show ip bgp prefix-list <name>***

## Distance

- ***distance <dist> <source IP> <source mask> [<acl>]***
  Set distance for specific prefixes received from specific peer
- ***distance bgp <ext> <int> <local/backdoor>***
  Set distance for all prefixes

## Path Filters

- ***ip as-path access-list <id> permit|deny <regexp>***
- ***neighbor <ip> filter-list <id> in|out***
- ***show ip bgp filer-list <id>***
- ***show ip bgp regexp <regexp>***

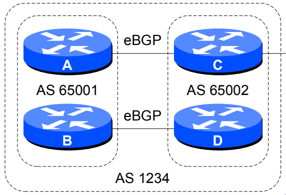
## Distribute List

- ***access-list <id> permit host <net> host <mask>***
  Exact match for the prefix (specific network with specific netmask)
- ***access-list <id> permit <net> <rev-mask-for-net> <mask> <rev-mask-for-mask>***
  Alternate solutiuon for prefix-lists. Manipulating network and netmask wildcards, LE/GE -like features can be implemented using ACLs. Works only for BGP.

## Route-Map

- Policy-list - macro
  - ***ip policy-list <name> permit|deny***
    ***match …***
    ***route-map <name> permit|deny***
    ***match policy-list <name>***
- If RM entry contains only set clauses they are all executed and no other RM entries are evaluated
- ***neighbor <ip> route-map <name> in|out***
- ***show ip bgp route-map <name>***
- ***set ip next-hop <ip> …***
  Better granularity than next-hop-self (which applies to all routes)
- ***set ip next-hop peer-address***
  If used in „out” route-map then local interface's IP is used as a next hop, if used in „in” route-map then peer's IP is used as a next-hop.

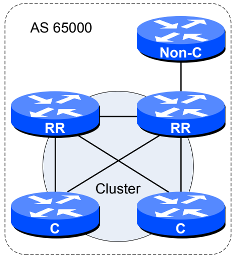
# BGP Scalability



## Confederation

- As loop prevention AS_CONFED_SEQUENCE and AS_CONFED_SET is maintained. Each AS adds own sub-AS to path. {65001 65002}
- When update is sent to external peer the AS_CONFED_SEQUENCE and AS_CONFED_SET information is stripped from the AS_PATH attribute, and the confederation ID is prepended to the AS_PATH
- ***router bgp <id>*** (private AS)
  ***bgp confederation identifier <id>*** (real AS)
  ***bgp confederation peers <as> <as>*** (sub-ASes)
- Centralized design recommended
- NEXT_HOP, MED, LOCAL_PREF left untouched between sub-ASes, common IGP required
- eBGP between sub-Ases (Preference: ext eBGP -> confed ext eBGP -> iBGP)
- Advertisement follows simple eBGP and iBGP rules

## Route-Reflectors



- CLLUSTER_LIST updated by RR with CLUSTER_ID (usualu router ID) when RR sends route from client to non-client. Loop avoidance
- ORIGINATOR_ID added by RR in Update sourced by a client. RR will not send update to a peer the same as originator-id. Router which is an originator will drop update with originator-id set to own. Loop avoidance.
- Route from non-client reflect to clients and eBGP peers only
- Route from eBGP reflect to clients and non-clients
- Route from client reflect to non-clients, clients and eBGP peers
- Route-reflector in different cluster is a non-client for local route-reflecotr
- ***neighbor <ip> route-reflector-client***
  Define client on RR. Client is not aware of being a client
- ***bgp cluster-id <id>***
  Set if more than one RR in a cluster – not recommended. Cluster is a set of route reflectors and its clients. Clusters may overlap. If not set, it is a router ID
- connections between clusters must be made between the route reflectors, not between clients, because clients do not examine the CLUSTER_LIST
- ***no bgp client-to-client reflection***
  When the clients are fully meshed, the route reflector is configured so that it does not reflect routes from one client to another
- RR can be implemented hierarchicaly
- Physical path should follow RR-to-Client path to avoid blackholing and loops

# BGP Stability

## Soft Reconfig

- Peer's table version is reset to 0, next update interval local router sends whole BGP table.
- ***neighbor <ip> soft-reconfiguration inbound***
- ***clear ip bgp <id> soft in|out***

## ORF

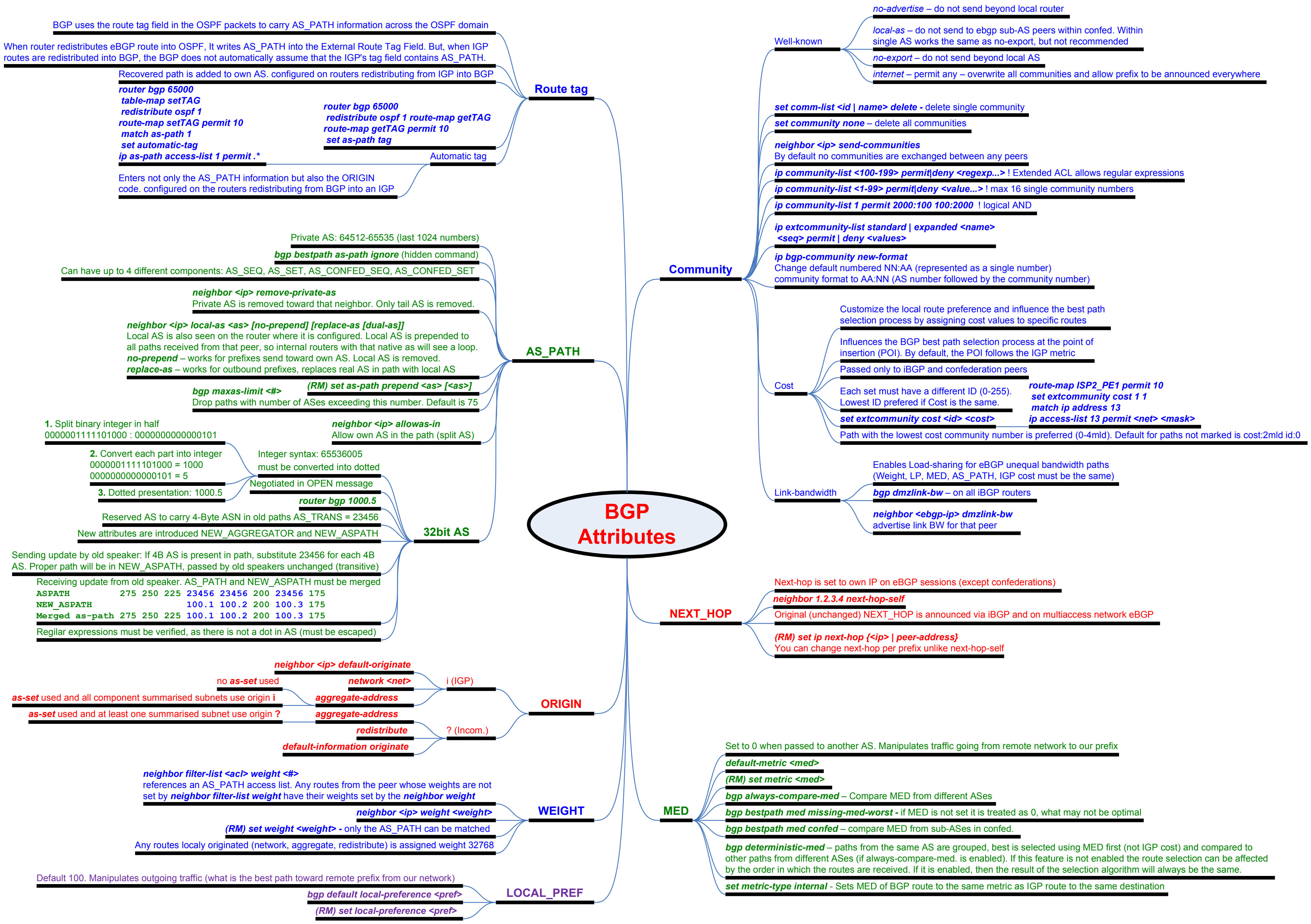
- Only for individual peers. Multicast not supported
- Requires prefix-list configuration (the only method supported)
- BGP speaker can install the inbound prefix list filter to the remote peer as an outbound filter
- ***neighbor <ip> capability orf prefix-list send|receive|both***
  ***neighbor <ip> prefix-list FILTER in***
- ***show ip bgp neighbor 10.1.1.2 received prefix-filter***
- ***clear ip bgp <ip> in [prefix-filter] -*** trigger route refresh

## Route Refresh

- Replacement for soft-reconfiguration; Negotiated when session is established
- Dynamicaly request Adj-RIP-out from peer
- ***clear ip bgp <id> in***

## Dampening

- Penalty added to specific path, not prefix. Flap means down and up. If path goes only down it is not a flap.
- Max Penalty = Reuse Limit * 2 * (Max Suppress Time / Half Life)
- Half-life: 15min; Reuse: 750; Suppress: 2000; Max: 4xHalf-life; Penalty: 1000
- Penalty is reduced every 5 sec in a way that after 15 min is half
- ***bgp dampening {[route-map <name>]} | {[<half-life> <reuse> <supp> <max-supp>]}***
- ***set dampening (route-map)***
- Flap history is cleared when penalty drops below half of reuse-limit
- ***clear ip bgp dampening***
- ***clear ip bgp <peer-ip> flap-statistics***



# BGP Attributes

## Route tag

BGP uses the route tag field in the OSPF packets to carry AS_PATH information across the OSPF domain

When router redistributes eBGP route into OSPF, It writes AS_PATH into the External Route Tag Field. But, when IGP routes are redistributed into BGP, the BGP does not automatically assume that the IGP's tag field contains AS_PATH.

Recovered path is added to own AS. configured on routers redistributing from IGP into BGP

```
router bgp 65000
 redistribute ospf 1 route-map getTAG
route-map getTAG permit 10
 set as-path tag
```

Automatic tag

```
router bgp 65000
 table-map setTAG
 redistribute ospf 1
route-map setTAG permit 10
 match as-path 1
 set automatic-tag
ip as-path access-list 1 permit .*
```

Enters not only the AS_PATH information but also the ORIGIN code. configured on the routers redistributing from BGP into an IGP

## AS_PATH

Private AS: 64512-65535 (last 1024 numbers)

***bgp bestpath as-path ignore*** (hidden command)

Can have up to 4 different components: AS_SEQ, AS_SET, AS_CONFED_SEQ, AS_CONFED_SET

***neighbor <ip> remove-private-as***
Private AS is removed toward that neighbor. Only tail AS is removed.

***neighbor <ip> local-as <as> [no-prepend] [replace-as [dual-as]]***
Local AS is also seen on the router where it is configured. Local AS is prepended to all paths received from that peer, so internal routers with that native as will see a loop.
***no-prepend*** – works for prefixes send toward own AS. Local AS is removed.
***replace-as*** – works for outbound prefixes, replaces real AS in path with local AS

***(RM) set as-path prepend <as> [<as>]***

***bgp maxas-limit <#>***
Drop paths with number of ASes exceeding this number. Default is 75

***neighbor <ip> allowas-in***
Allow own AS in the path (split AS)

## 32bit AS

Integer syntax: 65536005 must be converted into dotted

**1.** Split binary integer in half
0000001111101000 : 0000000000000101

**2.** Convert each part into integer
0000001111101000 = 1000
0000000000000101 = 5

**3.** Dotted presentation: 1000.5

Negotiated in OPEN message

***router bgp 1000.5***

Reserved AS to carry 4-Byte ASN in old paths AS_TRANS = 23456

New attributes are introduced NEW_AGGREGATOR and NEW_ASPATH

Sending update by old speaker: If 4B AS is present in path, substitute 23456 for each 4B AS. Proper path will be in NEW_ASPATH, passed by old speakers unchanged (transitive)

Receiving update from old speaker. AS_PATH and NEW_ASPATH must be merged

```
ASPATH          275 250 225 23456 23456 200 23456 175
NEW_ASPATH                  100.1 100.2 200 100.3 175
Merged as-path  275 250 225 100.1 100.2 200 100.3 175
```

Regilar expressions must be verified, as there is not a dot in AS (must be escaped)

## ORIGIN

i (IGP)
- ***neighbor <ip> default-originate***
- ***network <net>***
- ***aggregate-address*** – no ***as-set*** used; ***as-set*** used and all component summarised subnets use origin **i**

? (Incom.)
- ***aggregate-address*** – ***as-set*** used and at least one summarised subnet use origin **?**
- ***redistribute***
- ***default-information originate***

## WEIGHT

***neighbor filter-list <acl> weight <#>***
references an AS_PATH access list. Any routes from the peer whose weights are not set by ***neighbor filter-list weight*** have their weights set by the ***neighbor weight***

***neighbor <ip> weight <weight>***

***(RM) set weight <weight> -*** only the AS_PATH can be matched

Any routes localy originated (network, aggregate, redistribute) is assigned weight 32768

## LOCAL_PREF

Default 100. Manipulates outgoing traffic (what is the best path toward remote prefix from our network)

***bgp default local-preference <pref>***

***(RM) set local-preference <pref>***

## Community

Well-known
- *no-advertise* – do not send beyond local router
- *local-as* – do not send to ebgp sub-AS peers within confed. Within single AS works the same as no-export, but not recommended
- *no-export* – do not send beyond local AS
- *internet* – permit any – overwrite all communities and allow prefix to be announced everywhere

***set comm-list <id | name> delete -*** delete single community

***set community none*** – delete all communities

***neighbor <ip> send-communities***
By default no communities are exchanged between any peers

***ip community-list <100-199> permit|deny <regexp...>*** ! Extended ACL allows regular expressions

***ip community-list <1-99> permit|deny <value...>*** ! max 16 single community numbers

***ip community-list 1 permit 2000:100 100:2000*** ! logical AND

***ip extcommunity-list standard | expanded <name>***
 ***<seq> permit | deny <values>***

***ip bgp-community new-format***
Change default numbered NN:AA (represented as a single number) community format to AA:NN (AS number followed by the community number)

Cost
- Customize the local route preference and influence the best path selection process by assigning cost values to specific routes
- Influences the BGP best path selection process at the point of insertion (POI). By default, the POI follows the IGP metric
- Passed only to iBGP and confederation peers
- Each set must have a different ID (0-255). Lowest ID prefered if Cost is the same.
- ***set extcommunity cost <id> <cost>***
- Path with the lowest cost community number is preferred (0-4mld). Default for paths not marked is cost:2mld id:0

```
route-map ISP2_PE1 permit 10
 set extcommunity cost 1 1
 match ip address 13
ip access-list 13 permit <net> <mask>
```

Link-bandwidth
- Enables Load-sharing for eBGP unequal bandwidth paths (Weight, LP, MED, AS_PATH, IGP cost must be the same)
- ***bgp dmzlink-bw*** – on all iBGP routers
- ***neighbor <ebgp-ip> dmzlink-bw***
  advertise link BW for that peer

## NEXT_HOP

Next-hop is set to own IP on eBGP sessions (except confederations)

***neighbor 1.2.3.4 next-hop-self***

Original (unchanged) NEXT_HOP is announced via iBGP and on multiaccess network eBGP

***(RM) set ip next-hop {<ip> | peer-address}***
You can change next-hop per prefix unlike next-hop-self

## MED

Set to 0 when passed to another AS. Manipulates traffic going from remote network to our prefix

***default-metric <med>***

***(RM) set metric <med>***

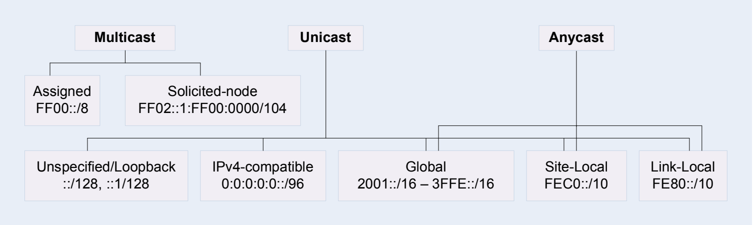
***bgp always-compare-med*** – Compare MED from different ASes

***bgp bestpath med missing-med-worst -*** if MED is not set it is treated as 0, what may not be optimal

***bgp bestpath med confed*** – compare MED from sub-ASes in confed.

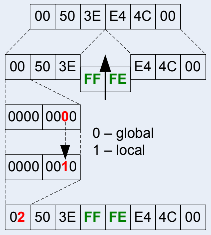
***bgp deterministic-med*** – paths from the same AS are grouped, best is selected using MED first (not IGP cost) and compared to other paths from different ASes (if always-compare-med. is enabled). If this feature is not enabled the route selection can be affected by the order in which the routes are received. If it is enabled, then the result of the selection algorithm will always be the same.

***set metric-type internal*** - Sets MED of BGP route to the same metric as IGP route to the same destination



## EUI-64

48bit MAC => 64bit EUI conversion

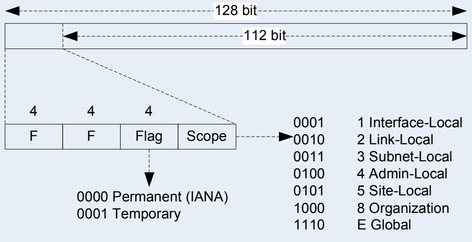


**Step 1**
Insert FFFE
in the middle

**Step 2**
7th most significant
bit flipped (not set to
1, but always flipped)

## Multicast FF00::/8

No TTL. Scoping in address. Src address can never be Mcast.



| | |
|---|---|
| 0001 | 1 Interface-Local |
| 0010 | 2 Link-Local |
| 0011 | 3 Subnet-Local |
| 0100 | 4 Admin-Local |
| 0101 | 5 Site-Local |
| 1000 | 8 Organization |
| 1110 | E Global |

| | |
|---|---|
| FF02::1 | All Nodes |
| FF02::2 | All Routers |
| FF02::5 | OSPFv3 Routers |
| FF02::6 | OSPFv3 DRs |
| FF02::9 | RIPng Routers |
| FF02::A | EIGRP Routers |
| FF02::B | Mobile Agents |
| FF02::C | DHCP Servers/Relay |
| FF02::D | All PIM Routers |

| | |
|---|---|
| **::/128** | **Unspecified** |
| **::1/128** | **Loopback** |
| **::/0** | **Default** |

## Aggregatable-Global

## 2000::/3 – 3FFF:FFFF...FFFF

/48 provider + /16 site + EUI-64

| | |
|---|---|
| 2001::/16 | IPv6 Internet |
| 2002::/16 | 6to4 transition mechanisms |
| 2003::/16 | Unassigned |
| 3FFD::/16 | Unassigned |
| 3FFE::/16 | 6bone |

## Link-Local

## FE80::/10 + EUI-64

## Site-Local (Obsoleted)

## FEC0::/10 + EUI-64

## Unique Local (ULA)

## FC00::/7 + EUI-64

## Solicited node Mcast:

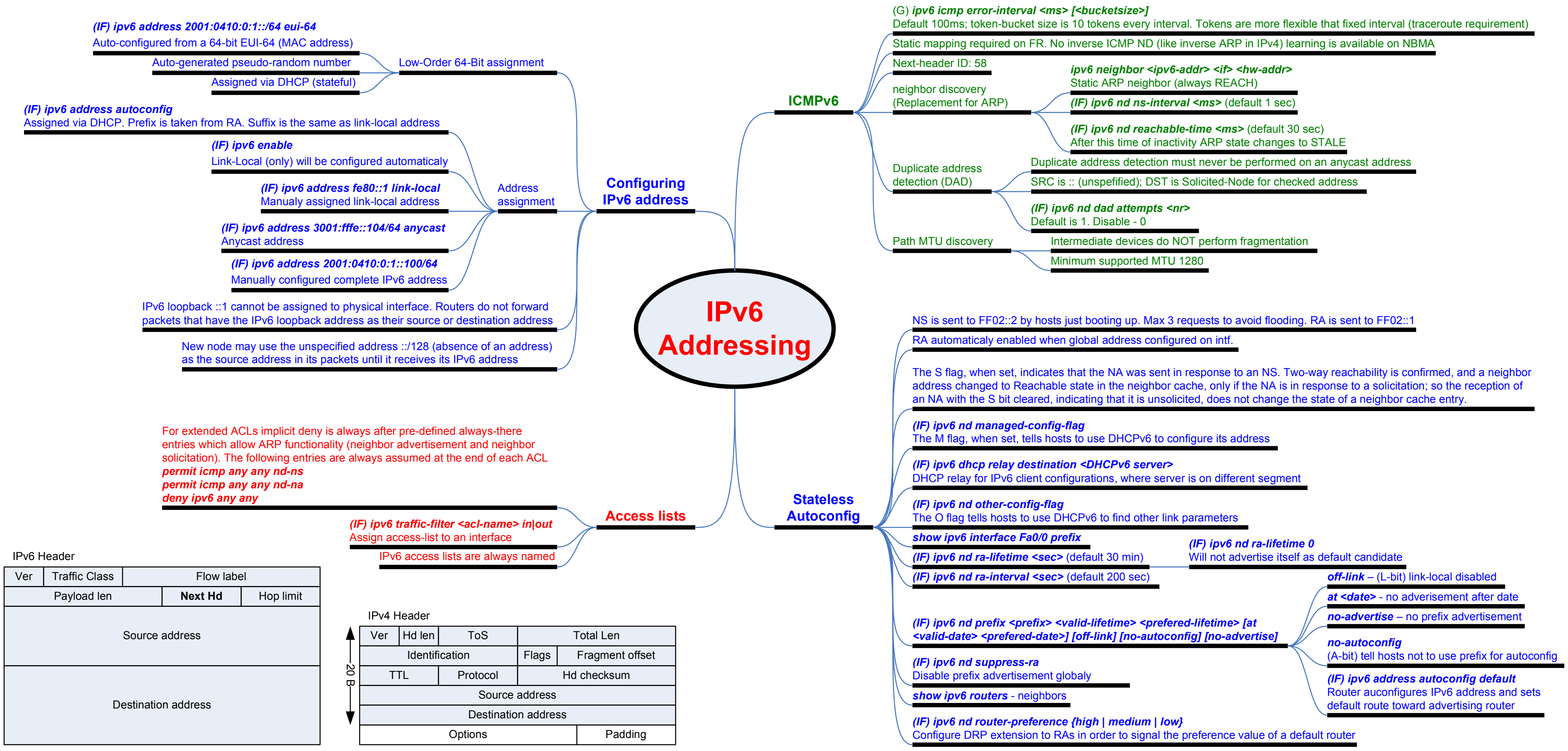
## FF02::1:FFxx::xxxx/104 + LO 24bit uncst

Automaticaly created for each unicast or anycast. „ARP", DAD.

## Multicast => MAC

## 33:33 + low-order 32 bit

FF02::1 => 33:33:00:00:00:01 MAC



# IPv6 Addressing

## Configuring IPv6 address

Low-Order 64-Bit assignment
- ***(IF) ipv6 address 2001:0410:0:1::/64 eui-64***
  Auto-configured from a 64-bit EUI-64 (MAC address)
- Auto-generated pseudo-random number
- Assigned via DHCP (stateful)

Address assignment
- ***(IF) ipv6 address autoconfig***
  Assigned via DHCP. Prefix is taken from RA. Suffix is the same as link-local address
- ***(IF) ipv6 enable***
  Link-Local (only) will be configured automaticaly
- ***(IF) ipv6 address fe80::1 link-local***
  Manualy assigned link-local address
- ***(IF) ipv6 address 3001:fffe::104/64 anycast***
  Anycast address
- ***(IF) ipv6 address 2001:0410:0:1::100/64***
  Manually configured complete IPv6 address

IPv6 loopback ::1 cannot be assigned to physical interface. Routers do not forward packets that have the IPv6 loopback address as their source or destination address

New node may use the unspecified address ::/128 (absence of an address) as the source address in its packets until it receives its IPv6 address

## Access lists

For extended ACLs implicit deny is always after pre-defined always-there entries which allow ARP functionality (neighbor advertisement and neighbor solicitation). The following entries are always assumed at the end of each ACL
***permit icmp any any nd-ns***
***permit icmp any any nd-na***
***deny ipv6 any any***

***(IF) ipv6 traffic-filter <acl-name> in|out***
Assign access-list to an interface

IPv6 access lists are always named

## ICMPv6

(G) ***ipv6 icmp error-interval <ms> [<bucketsize>]***
Default 100ms; token-bucket size is 10 tokens every interval. Tokens are more flexible that fixed interval (traceroute requirement)

Static mapping required on FR. No inverse ICMP ND (like inverse ARP in IPv4) learning is available on NBMA

Next-header ID: 58

neighbor discovery (Replacement for ARP)
- ***ipv6 neighbor <ipv6-addr> <if> <hw-addr>***
  Static ARP neighbor (always REACH)
- ***(IF) ipv6 nd ns-interval <ms>*** (default 1 sec)
- ***(IF) ipv6 nd reachable-time <ms>*** (default 30 sec)
  After this time of inactivity ARP state changes to STALE

Duplicate address detection (DAD)
- Duplicate address detection must never be performed on an anycast address
- SRC is :: (unspefified); DST is Solicited-Node for checked address
- ***(IF) ipv6 nd dad attempts <nr>***
  Default is 1. Disable - 0

Path MTU discovery
- Intermediate devices do NOT perform fragmentation
- Minimum supported MTU 1280

## Stateless Autoconfig

NS is sent to FF02::2 by hosts just booting up. Max 3 requests to avoid flooding. RA is sent to FF02::1

RA automaticaly enabled when global address configured on intf.

The S flag, when set, indicates that the NA was sent in response to an NS. Two-way reachability is confirmed, and a neighbor address changed to Reachable state in the neighbor cache, only if the NA is in response to a solicitation; so the reception of an NA with the S bit cleared, indicating that it is unsolicited, does not change the state of a neighbor cache entry.

***(IF) ipv6 nd managed-config-flag***
The M flag, when set, tells hosts to use DHCPv6 to configure its address

***(IF) ipv6 dhcp relay destination <DHCPv6 server>***
DHCP relay for IPv6 client configurations, where server is on different segment

***(IF) ipv6 nd other-config-flag***
The O flag tells hosts to use DHCPv6 to find other link parameters

***show ipv6 interface Fa0/0 prefix***

***(IF) ipv6 nd ra-lifetime <sec>*** (default 30 min)
- ***(IF) ipv6 nd ra-lifetime 0***
  Will not advertise itself as default candidate

***(IF) ipv6 nd ra-interval <sec>*** (default 200 sec)

***(IF) ipv6 nd prefix <prefix> <valid-lifetime> <prefered-lifetime> [at <valid-date> <prefered-date>] [off-link] [no-autoconfig] [no-advertise]***
- ***off-link*** – (L-bit) link-local disabled
- ***at <date>*** - no adverisement after date
- ***no-advertise*** – no prefix advertisement
- ***no-autoconfig***
  (A-bit) tell hosts not to use prefix for autoconfig
- ***(IF) ipv6 address autoconfig default***
  Router auconfigures IPv6 address and sets default route toward advertising router

***(IF) ipv6 nd suppress-ra***
Disable prefix advertisement globaly

***show ipv6 routers*** - neighbors

***(IF) ipv6 nd router-preference {high | medium | low}***
Configure DRP extension to RAs in order to signal the preference value of a default router

IPv6 Header

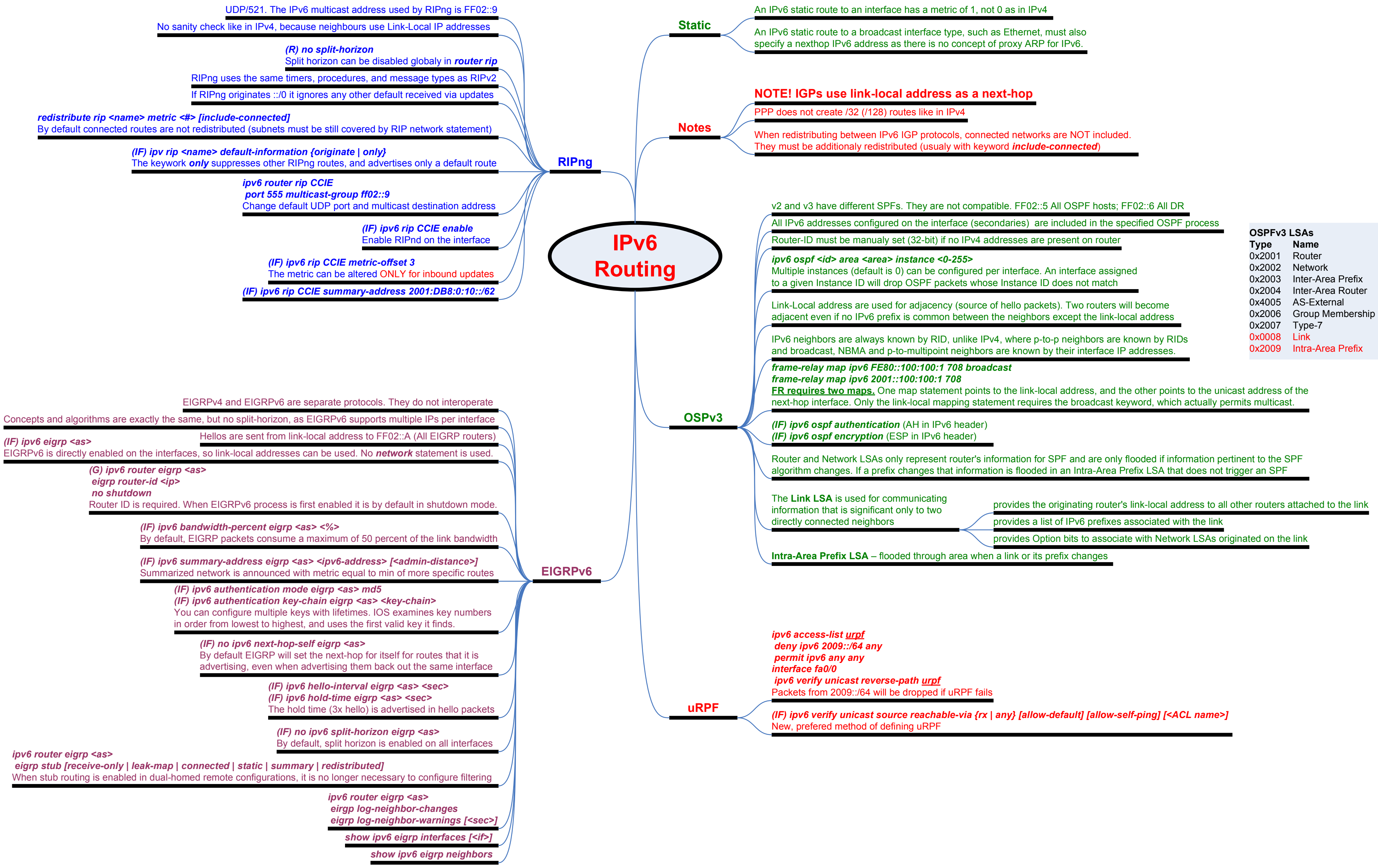
40 B

| | | |
|---|---|---|
| Ver | Traffic Class | Flow label |
| Payload len | **Next Hd** | Hop limit |
| Source address | | |
| Destination address | | |

IPv4 Header

20 B

| | | | |
|---|---|---|---|
| Ver | Hd len | ToS | Total Len |
| Identification | | Flags | Fragment offset |
| TTL | | Protocol | Hd checksum |
| Source address | | | |
| Destination address | | | |
| Options | | | Padding |



# IPv6 Routing

## RIPng

- UDP/521. The IPv6 multicast address used by RIPng is FF02::9
- No sanity check like in IPv4, because neighbours use Link-Local IP addresses
- ***(R) no split-horizon***
  Split horizon can be disabled globaly in ***router rip***
- RIPng uses the same timers, procedures, and message types as RIPv2
- If RIPng originates ::/0 it ignores any other default received via updates
- ***redistribute rip <name> metric <#> [include-connected]***
  By default connected routes are not redistributed (subnets must be still covered by RIP network statement)
- ***(IF) ipv rip <name> default-information {originate | only}***
  The keywork ***only*** suppresses other RIPng routes, and advertises only a default route
- ***ipv6 router rip CCIE***
  ***port 555 multicast-group ff02::9***
  Change default UDP port and multicast destination address
- ***(IF) ipv6 rip CCIE enable***
  Enable RIPnd on the interface
- ***(IF) ipv6 rip CCIE metric-offset 3***
  The metric can be altered ONLY for inbound updates
- ***(IF) ipv6 rip CCIE summary-address 2001:DB8:0:10::/62***

## EIGRPv6

- EIGRPv4 and EIGRPv6 are separate protocols. They do not interoperate
- Concepts and algorithms are exactly the same, but no split-horizon, as EIGRPv6 supports multiple IPs per interface
- Hellos are sent from link-local address to FF02::A (All EIGRP routers)
- ***(IF) ipv6 eigrp <as>***
  EIGRPv6 is directly enabled on the interfaces, so link-local addresses can be used. No ***network*** statement is used.
- ***(G) ipv6 router eigrp <as>***
  ***eigrp router-id <ip>***
  ***no shutdown***
  Router ID is required. When EIGRPv6 process is first enabled it is by default in shutdown mode.
- ***(IF) ipv6 bandwidth-percent eigrp <as> <%>***
  By default, EIGRP packets consume a maximum of 50 percent of the link bandwidth
- ***(IF) ipv6 summary-address eigrp <as> <ipv6-address> [<admin-distance>]***
  Summarized network is announced with metric equal to min of more specific routes
- ***(IF) ipv6 authentication mode eigrp <as> md5***
  ***(IF) ipv6 authentication key-chain eigrp <as> <key-chain>***
  You can configure multiple keys with lifetimes. IOS examines key numbers in order from lowest to highest, and uses the first valid key it finds.
- ***(IF) no ipv6 next-hop-self eigrp <as>***
  By default EIGRP will set the next-hop for itself for routes that it is advertising, even when advertising them back out the same interface
- ***(IF) ipv6 hello-interval eigrp <as> <sec>***
  ***(IF) ipv6 hold-time eigrp <as> <sec>***
  The hold time (3x hello) is advertised in hello packets
- ***(IF) no ipv6 split-horizon eigrp <as>***
  By default, split horizon is enabled on all interfaces
- ***ipv6 router eigrp <as>***
  ***eigrp stub [receive-only | leak-map | connected | static | summary | redistributed]***
  When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering
- ***ipv6 router eigrp <as>***
  ***eigrp log-neighbor-changes***
  ***eigrp log-neighbor-warnings [<sec>]***
- ***show ipv6 eigrp interfaces [<if>]***
- ***show ipv6 eigrp neighbors***

## Static

- An IPv6 static route to an interface has a metric of 1, not 0 as in IPv4
- An IPv6 static route to a broadcast interface type, such as Ethernet, must also specify a nexthop IPv6 address as there is no concept of proxy ARP for IPv6.

## Notes

- **NOTE! IGPs use link-local address as a next-hop**
- PPP does not create /32 (/128) routes like in IPv4
- When redistributing between IPv6 IGP protocols, connected networks are NOT included. They must be additionaly redistributed (usualy with keyword ***include-connected***)

## OSPv3

- v2 and v3 have different SPFs. They are not compatible. FF02::5 All OSPF hosts; FF02::6 All DR
- All IPv6 addresses configured on the interface (secondaries) are included in the specified OSPF process
- Router-ID must be manualy set (32-bit) if no IPv4 addresses are present on router
- ***ipv6 ospf <id> area <area> instance <0-255>***
  Multiple instances (default is 0) can be configured per interface. An interface assigned to a given Instance ID will drop OSPF packets whose Instance ID does not match
- Link-Local address are used for adjacency (source of hello packets). Two routers will become adjacent even if no IPv6 prefix is common between the neighbors except the link-local address
- IPv6 neighbors are always known by RID, unlike IPv4, where p-to-p neighbors are known by RIDs and broadcast, NBMA and p-to-multipoint neighbors are known by their interface IP addresses.
- ***frame-relay map ipv6 FE80::100:100:1 708 broadcast***
  ***frame-relay map ipv6 2001::100:100:1 708***
  **FR requires two maps.** One map statement points to the link-local address, and the other points to the unicast address of the next-hop interface. Only the link-local mapping statement requires the broadcast keyword, which actually permits multicast.
- ***(IF) ipv6 ospf authentication*** (AH in IPv6 header)
  ***(IF) ipv6 ospf encryption*** (ESP in IPv6 header)
- Router and Network LSAs only represent router's information for SPF and are only flooded if information pertinent to the SPF algorithm changes. If a prefix changes that information is flooded in an Intra-Area Prefix LSA that does not trigger an SPF
- The **Link LSA** is used for communicating information that is significant only to two directly connected neighbors
  - provides the originating router's link-local address to all other routers attached to the link
  - provides a list of IPv6 prefixes associated with the link
  - provides Option bits to associate with Network LSAs originated on the link
- **Intra-Area Prefix LSA** – flooded through area when a link or its prefix changes

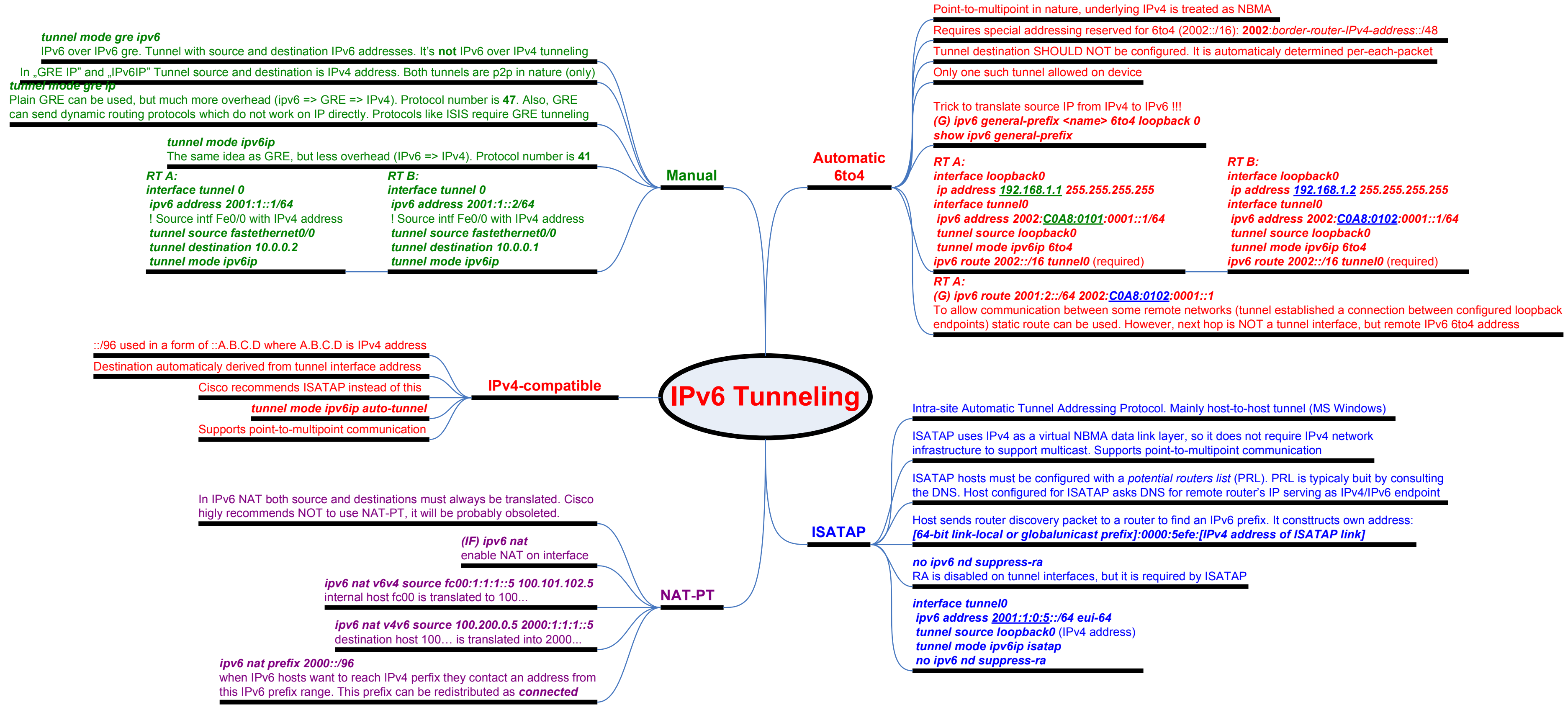
**OSPFv3 LSAs**

| Type | Name |
|---|---|
| 0x2001 | Router |
| 0x2002 | Network |
| 0x2003 | Inter-Area Prefix |
| 0x2004 | Inter-Area Router |
| 0x4005 | AS-External |
| 0x2006 | Group Membership |
| 0x2007 | Type-7 |
| 0x0008 | Link |
| 0x2009 | Intra-Area Prefix |

## uRPF

- ***ipv6 access-list urpf***
  ***deny ipv6 2009::/64 any***
  ***permit ipv6 any any***
  ***interface fa0/0***
  ***ipv6 verify unicast reverse-path urpf***
  Packets from 2009::/64 will be dropped if uRPF fails
- ***(IF) ipv6 verify unicast source reachable-via {rx | any} [allow-default] [allow-self-ping] [<ACL name>]***
  New, prefered method of defining uRPF

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



# IPv6 Tunneling

## Manual

***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 mode gre ip***
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 IP directly. Protocols like ISIS require GRE tunneling

***tunnel mode ipv6ip***
The same idea as GRE, but less overhead (IPv6 => IPv4). Protocol number is **41**

```
RT A:
interface tunnel 0
 ipv6 address 2001:1::1/64
 ! Source intf Fe0/0 with IPv4 address
 tunnel source fastethernet0/0
 tunnel destination 10.0.0.2
 tunnel mode ipv6ip
```

```
RT B:
interface tunnel 0
 ipv6 address 2001:1::2/64
 ! Source intf Fe0/0 with IPv4 address
 tunnel source fastethernet0/0
 tunnel destination 10.0.0.1
 tunnel mode ipv6ip
```

## Automatic 6to4

Point-to-multipoint in nature, underlying IPv4 is treated as NBMA

Requires special addressing reserved for 6to4 (2002::/16): **2002**:*border-router-IPv4-address*::/48

Tunnel destination SHOULD NOT be configured. It is automaticaly 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***

```
RT A:
interface loopback0
 ip address 192.168.1.1 255.255.255.255
interface tunnel0
 ipv6 address 2002:C0A8:0101:0001::1/64
 tunnel source loopback0
 tunnel mode ipv6ip 6to4
ipv6 route 2002::/16 tunnel0 (required)
```

```
RT B:
interface loopback0
 ip address 192.168.1.2 255.255.255.255
interface tunnel0
 ipv6 address 2002:C0A8:0102:0001::1/64
 tunnel source loopback0
 tunnel mode ipv6ip 6to4
ipv6 route 2002::/16 tunnel0 (required)
```

***RT A:***
***(G) ipv6 route 2001:2::/64 2002:C0A8:0102:0001::1***
To allow communication between some remote networks (tunnel established a connection between configured loopback endpoints) static route can be used. However, next hop is NOT a tunnel interface, but remote IPv6 6to4 address

## IPv4-compatible

::/96 used in a form of ::A.B.C.D where A.B.C.D is IPv4 address

Destination automaticaly derived from tunnel interface address

Cisco recommends ISATAP instead of this

***tunnel mode ipv6ip auto-tunnel***

Supports point-to-multipoint communication

## ISATAP

Intra-site Automatic Tunnel Addressing Protocol. Mainly host-to-host tunnel (MS Windows)

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 typicaly buit by consulting the DNS. Host configured for ISATAP asks DNS for remote router's IP serving as IPv4/IPv6 endpoint

Host sends router discovery packet to a router to find an IPv6 prefix. It consttructs own address:
***[64-bit link-local or globalunicast prefix]:0000:5efe:[IPv4 address of ISATAP link]***

***no ipv6 nd suppress-ra***
RA is disabled on tunnel interfaces, but it is required by ISATAP

```
interface tunnel0
 ipv6 address 2001:1:0:5::/64 eui-64
 tunnel source loopback0 (IPv4 address)
 tunnel mode ipv6ip isatap
 no ipv6 nd suppress-ra
```

## NAT-PT

In IPv6 NAT both source and destinations must always be translated. Cisco higly recommends NOT to use NAT-PT, it will be probably obsoleted.

***(IF) ipv6 nat***
enable NAT on interface

***ipv6 nat v6v4 source fc00:1:1:1::5 100.101.102.5***
internal host fc00 is translated to 100...

***ipv6 nat v4v6 source 100.200.0.5 2000:1:1:1::5***
destination host 100… is translated into 2000...

***ipv6 nat prefix 2000::/96***
when IPv6 hosts want to reach IPv4 perfix they contact an address from this IPv6 prefix range. This prefix can be redistributed as ***connected***

**224.0.0.0 – 239.255.255.255 (1110)**

224.0.0.0/24 – Link local
- .1 All hosts
- .2 All routers
- .4 DVMRP hosts
- .5 OSPF routers
- .6 OSPF DR
- .9 RIPv2
- .10 EIGRP routers
- .13 PIM routers
- .14 RSVP
- .15 All CBT routers
- .18 VRRP
- .22 IGMPv3

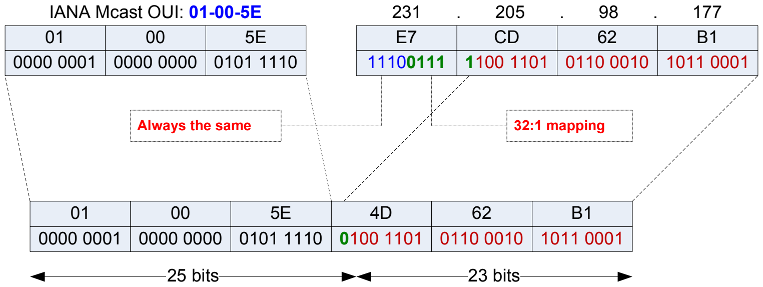
224.0.1.0/24 – IANA assigned
- .39 RP-Announce
- .40 RP-Discovery

232.0.0.0/8 – SSM

233.0.0.0/8 – GLOP (public AS to Mcast)

AS42123 => A4|8B => 164|139
233.164.139.0/24

239.0.0.0/8 – Administratively scoped



IANA Mcast OUI: **01-00-5E**

| 01 | 00 | 5E |
|---|---|---|
| 0000 0001 | 0000 0000 | 0101 1110 |

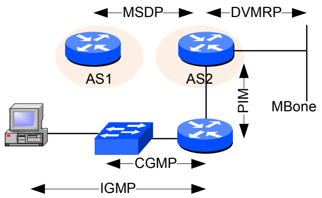
231 . 205 . 98 . 177

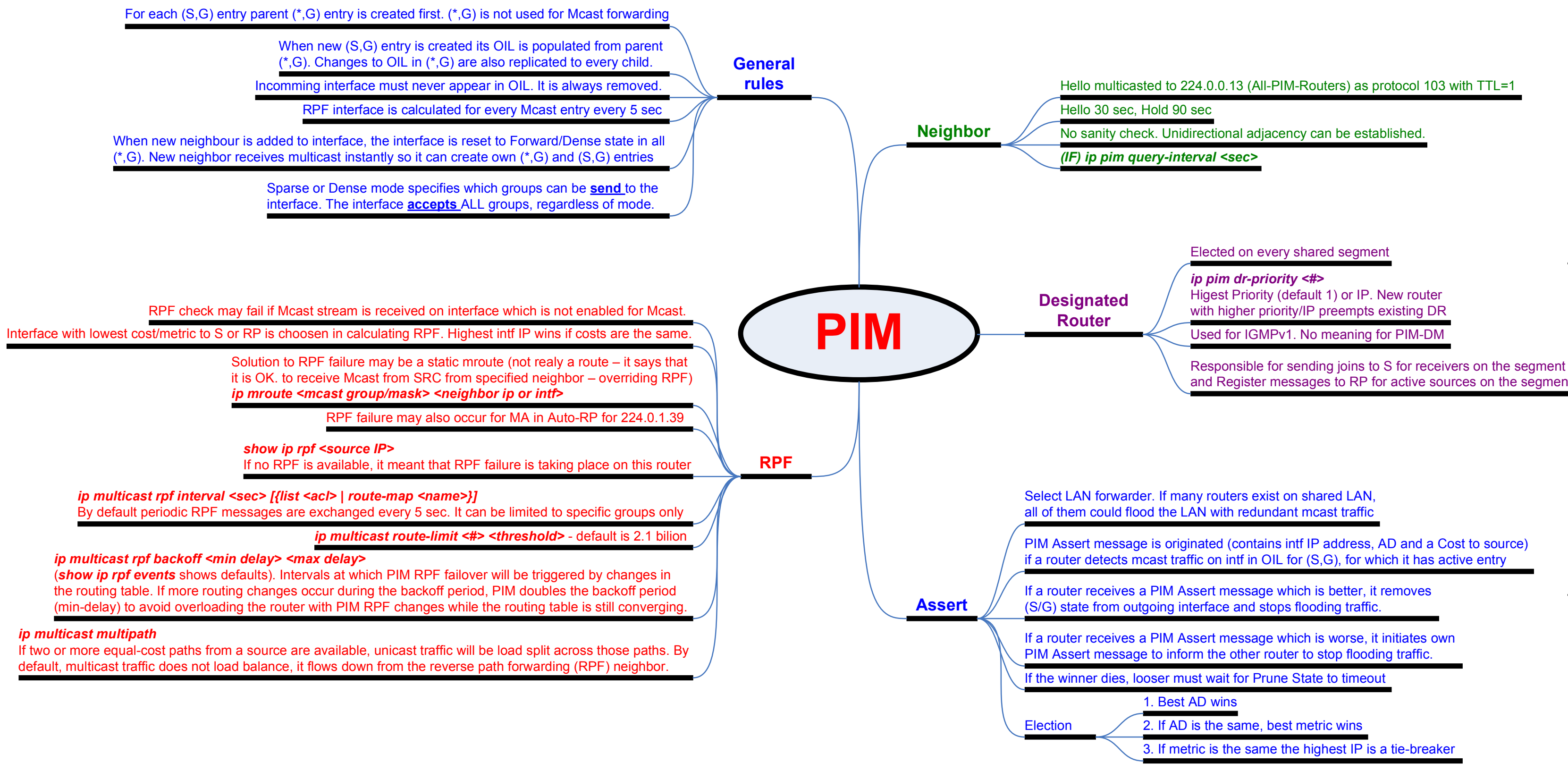
| E7 | CD | 62 | B1 |
|---|---|---|---|
| 11100111 | 1100 1101 | 0110 0010 | 1011 0001 |

**Always the same**

**32:1 mapping**

| 01 | 00 | 5E | 4D | 62 | B1 |
|---|---|---|---|---|---|
| 0000 0001 | 0000 0000 | 0101 1110 | 0100 1101 | 0110 0010 | 1011 0001 |





# PIM

## General rules

- For each (S,G) entry parent (*,G) entry is created first. (*,G) is not used for Mcast forwarding
- When new (S,G) entry is created its OIL is populated from parent (*,G). Changes to OIL in (*,G) are also replicated to every child.
- Incomming interface must never appear in OIL. It is always removed.
- RPF interface is calculated for every Mcast entry every 5 sec
- When 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
- Sparse or Dense mode specifies which groups can be **send** to the interface. The interface **accepts** ALL groups, regardless of mode.

## Neighbor

- Hello multicasted to 224.0.0.13 (All-PIM-Routers) as protocol 103 with TTL=1
- Hello 30 sec, Hold 90 sec
- No sanity check. Unidirectional adjacency can be established.
- ***(IF) ip pim query-interval <sec>***

## RPF

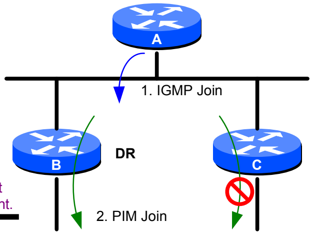
- RPF check may fail if Mcast stream is received on interface which is not enabled for Mcast.
- Interface with lowest cost/metric to S or RP is choosen in calculating RPF. Highest intf IP wins if costs are the same.
- Solution to RPF failure may be a static mroute (not realy a route – it says that it is OK. to receive Mcast from SRC from specified neighbor – overriding RPF)
  ***ip mroute <mcast group/mask> <neighbor ip or intf>***
- RPF failure may also occur for MA in Auto-RP for 224.0.1.39
- ***show ip rpf <source IP>***
  If no RPF is available, it meant that RPF failure is taking place on this router
- ***ip multicast rpf interval <sec> [{list <acl> | route-map <name>}]***
  By default periodic RPF messages are exchanged every 5 sec. It can be limited to specific groups only
- ***ip multicast route-limit <#> <threshold>*** - default is 2.1 bilion
- ***ip multicast rpf backoff <min delay> <max delay>***
  (***show ip rpf events*** shows defaults). Intervals at which PIM RPF failover will be triggered by changes in the routing table. If more routing changes occur during the backoff period, PIM doubles the backoff period (min-delay) to avoid overloading the router with PIM RPF changes while the routing table is still converging.
- ***ip multicast multipath***
  If two or more equal-cost paths from a source are available, unicast traffic will be load split across those paths. By default, multicast traffic does not load balance, it flows down from the reverse path forwarding (RPF) neighbor.

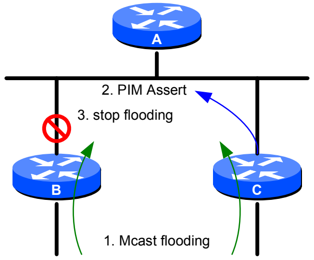
## Designated Router

- Elected on every shared segment
- ***ip pim dr-priority <#>***
  Higest Priority (default 1) or IP. New router with higher priority/IP preempts existing DR
- Used for IGMPv1. No meaning for PIM-DM
- Responsible for sending joins to S for receivers on the segment and Register messages to RP for active sources on the segment.

## Assert

- Select LAN forwarder. If many routers exist on shared LAN, all of them could flood the LAN with redundant mcast traffic
- PIM Assert message is originated (contains intf IP address, AD and a Cost to source) if a router detects mcast traffic on intf in OIL for (S,G), for which it has active entry
- 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.
- If the winner dies, looser must wait for Prune State to timeout
- Election
  - 1. Best AD wins
  - 2. If AD is the same, best metric wins
  - 3. If metric is the same the highest IP is a tie-breaker





# PIM-SM

## Source registration

- Register
  - Ucast to RP with encapsulated Mcast packets
  - RP joins SPT if receivers are present
- Register-Stop
  - S stops sending Ucast Registers
  - Sent by RP when starts receiving Mcast for (S,G) or automaticaly if no receivers are registered
  - 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

## STP Switchover

- 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)
- next packet check J-flag in (*,G) and if present sets J-flag in (S,G_ and joins SPT. (*,G) J-flag is cleared, and set back if next packet exceeds threshold
- 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

## Filtering

- Register filter
  - Defines which sources are allowed to register with RP. Configured only on RP.
  - ***ip pim accept-register {list <acl> | route-map <name>}***
  - Extended ACLs used for multicast filtering (any) is used as follow:
    ***access-list 100 permit <source ip> <wildcard> <group address> <wildcard>***
- Accept RP filter
  - Prevent unwanted RPs or mcast groups to became active in SM domain. Must be configured on every router.
  - ***ip pim accept-rp <rp-addr> [group-list <acl>]***
  - ***ip pim accept-rp 0.0.0.0*** (any)
  - ***ip pim accept-rp auto-rp*** (RP must be active in mapping)

## Rules

- Based on shared tree with a common root called randezvous point
- SM (*,G) entry is created as a result of Explicit Join. Either by directly connected IGMP join or by (*,G) join from downstream router
- Incoming interface of SM (*,G) always points to RP
- SM (S,G) is created (1) when received (S,G) Join/Prune message, (2) on last-hop-router when switched to SPT (3) on unexpected arrival of (S,G) trafic when no (*,G) exists, (4) on RP when Register is received
- Interface is added to OIL of SM (*,G) or (S,G) when (1) appropriate (*,G) or (S,G) Join is received via this intf, (2) directly connected members appears on that intf
- Interface is removed from OIL when (1) appropriate (*,G) or (S,G) Prune is received via this intf, (2) when interfaces expiration timer counts down to zero (3 min)
- Expiration timer is reset on (1) receiving appropriate (*,G) or (S,G) on intf, (2) receiving IGMP Report on that intf
- Routers will send (S,G) RP-bit Prune up to shared tree when RPF neighbour for (S,G) entryi different than (*,G) entry. RP-bit Prune is originated at the point where SPT and RPT diverge.
- RPF intf of SM (S,G) entry is calculated for S IP except for RP-bit when RP IP is used.
- **sparse-dense-mode**
  - Allows Auto-RP dense-mode groups 224.0.1.39 and 224.0.1.40 to be distributed while using sparse-mode groups.
  - ***no ip pim dm-fallback***
    Any group for which RP does not exists automaticaly switches by default back to DM

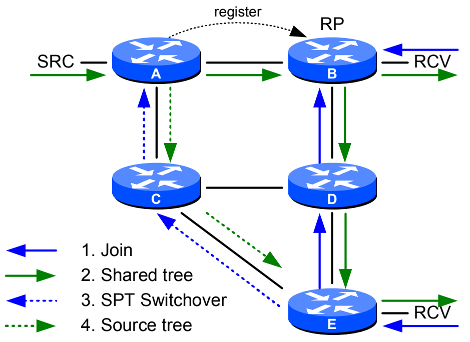
## NBMA

- ***(IF) ip pim nbma-mode***
- Works only for sparse-mode (relies on PIM Join)
- If Prune is received only specific entry is deleted
- Separate peers' next-hop is maintained in (*,G), and (S,G) OILs

## Randezvous Point

- Set manualy:
  ***ip pim rp-address <ip> [override]***
  - ***override*** option overrides Auto-RP
  - Can be used to prevent groups to switchover to DM when RP is down
- Auto-RIP (PIMv1, Cisco propietary)
- Bootstrap (PIMv2, standardized)
- ***show ip pim rp mapping***

### PIM sparse-mode operations



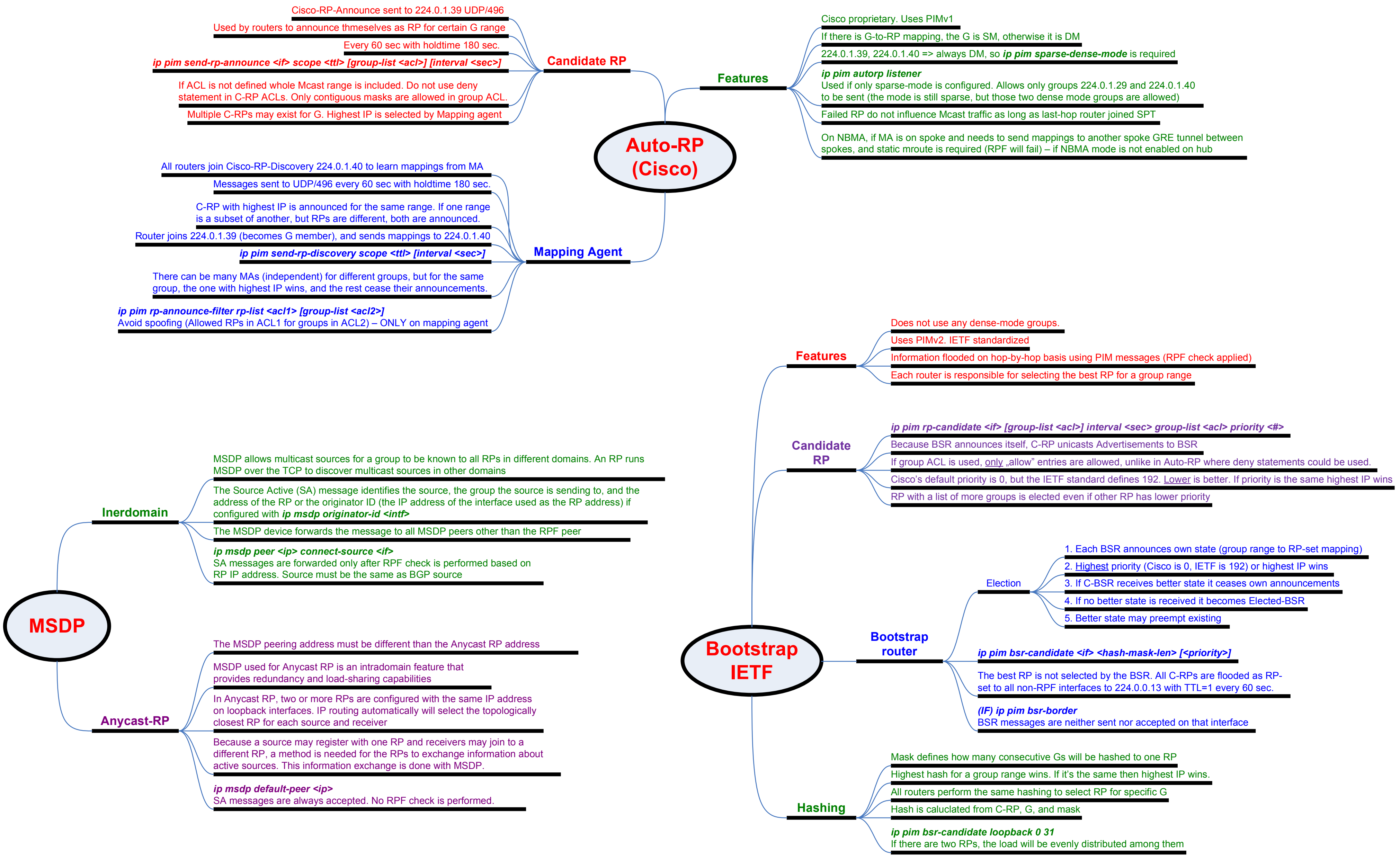
# PIM-DM

## Rules

- Based on source tree (shortest-path tree SPT)
- Flood and prune algorithm
- OIL list of (*,G) reflects interfaces where (1) neighbours 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

## Pruning

- Periodic (S,G) and (*,G) Joins are supressed.
- No (S,G) Prune messages are sent immediately, they timeout. Then, (S, G) Prunes are triggered by the arrival of (S, G) data packets (assuming S is still sending) for entry with P-flag set.
- (*,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 are deleted after 3-min timeout
- Prune-override – router waits 3 sec for Join from another router on shared LAN
- ***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.
- ***(IF) ip pim state-refresh origination-interval <sec>***



# Auto-RP (Cisco)

## Candidate RP

- Cisco-RP-Announce sent to 224.0.1.39 UDP/496
- Used by routers to announce thmeselves as RP for certain G range
- 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.
- Multiple C-RPs may exist for G. Highest IP is selected by Mapping agent

## Features

- Cisco proprietary. Uses PIMv1
- If there is G-to-RP mapping, the G is SM, otherwise it is DM
- 224.0.1.39, 224.0.1.40 => always DM, so ***ip pim sparse-dense-mode*** is required
- ***ip pim autorp listener***
  Used if only sparse-mode is configured. Allows only groups 224.0.1.29 and 224.0.1.40 to be sent (the mode is still sparse, but those two dense mode groups are allowed)
- Failed RP do not influence Mcast traffic as long as last-hop router joined SPT
- On NBMA, if MA is on spoke and needs to send mappings to another spoke GRE tunnel between spokes, and static mroute is required (RPF will fail) – if NBMA mode is not enabled on hub

## Mapping Agent

- All routers join Cisco-RP-Discovery 224.0.1.40 to learn mappings from MA
- Messages sent to UDP/496 every 60 sec with holdtime 180 sec.
- 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.
- Router joins 224.0.1.39 (becomes G member), and sends mappings to 224.0.1.40
- ***ip pim send-rp-discovery scope <ttl> [interval <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.
- ***ip pim rp-announce-filter rp-list <acl1> [group-list <acl2>]***
  Avoid spoofing (Allowed RPs in ACL1 for groups in ACL2) – ONLY on mapping agent

# Bootstrap IETF

## Features

- Does not use any dense-mode groups.
- Uses PIMv2. IETF standardized
- Information flooded on hop-by-hop basis using PIM messages (RPF check applied)
- Each router is responsible for selecting the best RP for a group range

## Candidate RP

- ***ip pim rp-candidate <if> [group-list <acl>] interval <sec> group-list <acl> priority <#>***
- Because BSR announces itself, C-RP unicasts Advertisements to BSR
- If group ACL is used, only „allow" entries are allowed, unlike in Auto-RP where deny statements could be used.
- Cisco's default priority is 0, but the IETF standard defines 192. Lower is better. If priority is the same highest IP wins
- RP with a list of more groups is elected even if other RP has lower priority

## Bootstrap router

- Election
  1. Each BSR announces own state (group range to RP-set mapping)
  2. Highest priority (Cisco is 0, IETF is 192) or highest IP wins
  3. If C-BSR receives better state it ceases own announcements
  4. If no better state is received it becomes Elected-BSR
  5. Better state may preempt existing
- ***ip pim bsr-candidate <if> <hash-mask-len> [<priority>]***
- The best RP is not selected by the BSR. All C-RPs are flooded as RP-set to all non-RPF interfaces to 224.0.0.13 with TTL=1 every 60 sec.
- ***(IF) ip pim bsr-border***
  BSR messages are neither sent nor accepted on that interface

## Hashing

- Mask defines how many consecutive Gs will be hashed to one RP
- 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
- Hash is caluclated from C-RP, G, and mask
- ***ip pim bsr-candidate loopback 0 31***
  If there are two RPs, the load will be evenly distributed among them

# MSDP

## Inerdomain

- MSDP allows multicast sources for a group to be known to all RPs in different domains. An RP runs MSDP over the TCP to discover multicast sources in other domains
- The Source Active (SA) message identifies the source, the group the source is sending to, and the address of the RP or the originator ID (the IP address of the interface used as the RP address) if configured with ***ip msdp originator-id <intf>***
- The MSDP device forwards the message to all MSDP peers other than the RPF peer
- ***ip msdp peer <ip> connect-source <if>***
  SA messages are forwarded only after RPF check is performed based on RP IP address. Source must be the same as BGP source

## Anycast-RP

- The MSDP peering address must be different than the Anycast RP address
- MSDP used for Anycast RP is an intradomain feature that provides redundancy and load-sharing capabilities
- In Anycast RP, two or more RPs are configured with the same IP address on loopback interfaces. IP routing automatically will select the topologically closest RP for each source and receiver
- Because a source may register with one RP and receivers may join to a different RP, a method is needed for the RPs to exchange information about active sources. This information exchange is done with MSDP.
- ***ip msdp default-peer <ip>***
  SA messages are always accepted. No RPF check is performed.

# IGMPv2

## Snooping

- 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
- **1.** router's Query is intercepted by CPU
  - **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.** One Report sent to router by CPU
- **1.** Host's Leave 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
- ***ip igmp snooping***
- ***ip igmp snooping vlan <id>***
- ***ip igmp snooping vlan <id> static <mac> interface <intf>***
- ***ip igmp snooping vlan vlan-id immediate-leave***
  Only if single receiver is present on the subnet

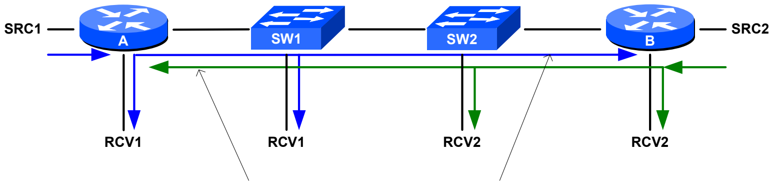
## CGMP

- L2 is examined by the router. Cisco proprietary; DST: 0100.0cdd.dddd
- Only router sends CGMP, and Switch only listens
- CAM entry is deleted if host's port chages state (STP change)
- 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)
- ***ip cgmp***
- **Join**
  - **1.** Host sends IGMP Join to R
  - **2.** R calculates Mcast MAC (GDA) from IP Mcast sent by host
  - **3.** R sends CGMP 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

| GDA | USA | J/L | Meaning |
|---|---|---|---|
| Mcast MAC | client MAC | Join | Add port to G |
| Mcast MAC | client MAC | Leave | Del port from G |
| 000...000 | router MAC | Join | Assign R port |
| 000...000 | router MAC | Leave | De-assign R port |
| Mcast MAC | 000...000 | Leave | Delete group |
| 000...000 | 000...000 | Leave | Delete all groups |

## RGMP

- Works well with IGMP snooping, but in addition helps to control ports on which routers are connected
- Router-Port Group Management Protocol. Cisco proprietary. If enabled CGMP is silently disabled
- RGMP is designed for switched Ethernet backbone networks running PIM sparse mode (PIM-SM) or sparse-dense mode
- RGMP enables a router to communicate to a switch the IP multicast group for which the router would like to receive or forward traffic
- ***(IF) ip rgmp***
- Messages sent only by router to 224.0.0.25
  - Hello every 30 sec – instructs the switch to restrict all multicast traffic on the interface from which the switch received the RGMP hello message
  - Join G – switch starts sending only G traffic via router port
  - Leave G – switch stops sending G traffic via router port
  - Bye – switch starts sending all groups traffic via router port (RGMP disabled)



Unnecessary flooding without RGMP

## Query

- Enabling a PIM on an interface enables IGMPv2
- Generic Q (0.0.0.0) to 224.0.0.1; Group-specific Q sent to G address
- 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
- Other Querier Present Interval = 255 (2x General Q Int + ½ of Q Response int)
- Timers
  - ***(IF) ip igmp query-interval <sec>***
    Default is 60 seconds. Automaticaly sets querier-timeout to 2x query int.
  - ***(IF) ip igmp querier-timeout <sec>***
    Backup querier becomes an active one if does not heare queries from the other router within this amount of time

## Report

- Join sent to G addr to which hosts wishes to join
- Leave sent to 224.0.0.2 (All routers)
- Report contains all groups to which host joined
- Timers
  - ***(IF) ip igmp query-max-response-time <1/10th of sec>***
    Max response int. 10 sec (tunable) in 1/10ths of sec. Time to wait for report supression on LAN segment. Excessive flooding on LAN is avoided.
  - ***(IF) ip igmp last-member-query-interval <sec>***
    Group-specific query interval. Query generated after receiving a leave from one host to see if there are other hosts in that group. Default is 1 sec.
  - ***(IF) ip igmp last-member-query-count <#>***
    Default is 2. Number of group-specific queries generated.
  - ***ip igmp immediate-leave group-list <acl>***
    If there is only one host connected to the LAN, the IGMP Leave for matched group causes mroute entry to be immediately deleted with no delays.

## Testing

- ***(IF) ip igmp join-group <group> [source <src IP>]***
  Pingable [only from specific source]
- ***(IF) ip igmp static-group***
  Non-pingable

## Filtering

- Switch
  - ***(IF) ip igmp filter <id>***
    ***ip igmp profile <id>***
    ***deny***
    ***range 224.1.1.1 224.1.1.50***
    You only define what is denied, the rest is allowed by default
  - ***(IF) ip igmp max-gr <#>***
    Limit number of groups to join on the interface
- Router
  - ***(IF) ip igmp access-group <name>***
    ***ip access-list standard <name>***
    ***deny 224.1.1.1***
    ***permit any***
    ACL can be also extended to limit specific hosts from joining groups
  - ***(G) ip igmp limit <#>***
    Configure a global limit on the number of mroute states created as a result of IGMP membership reports (IGMP joins).
  - ***(IF) ip igmp limit <#> [except <acl>]***
    If ACL is used, it Prevents groups from being counted against the interface limit. A standard ACL can be used to define the (*, G) state. An extended ACLs can be used to define the (S, G) state

# Mcast features

## MVR

- Multicast VLAN registration intercepts IGMP Joins
- Allows subscriber on a port to subscribe to a multicast stream on the network-wide multicast VLAN. Single multicast VLAN can be shared in the network while subscribers remain in separate VLANs
- Multicast routing and MVR cannot coexist on a switch
- ***(G) mvr*** – enable MVR
- ***mvr group <ip> [<count>]***
  Enbale MVR for a group and # of consecutive groups (max 256). Groups should not be aliasing (32:1 ratio)
- ***mvr mode dynamic -*** Default mode is compatible, which requires static IGMP snooping  entries
- ***mvr vlan <id> -*** Define which VLAN carries actual multicast traffic
- ***(IF) mvr type {source | receiver} -*** Define source and receiver interfaces

## Rate Limit

- ***ip multicast rate-limit {in | out} [group-list <acl>] [source-list <acl>] [<kbps>]***
- If limit speed is omited, the matched traffic is dropped

## Filtering

### TTL Threshold

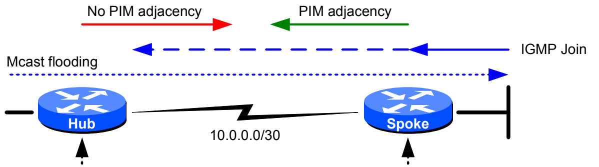
- ***(IF) ip multicast ttl-threshold <#>***
  By default all mcast enabled interfaces have TTL 0 – TTL in mcast packet must be higher than configured on interface

### Multicast boundary

- PIM Register messages cannot be filtered with this feature
- ***(IF) ip multicast boundry <acl> [filter-autorp]***
  ***access-list <acl> deny 224.0.1.39***
  ***access-list <acl> deny 224.0.1.40***
  ***access-list <acl> permit 224.0.0.0 15.255.255.255***
- 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.

## Stub router

- ***(IF) ip igmp helper-address <hub's WAN IP>***
  Configured on spoke's LAN interface. It forwards all IGMP messages to a Hub
- 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)
- ***(IF) 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.



```
interface serial 0/0
 ip pim sparse-dense-mode
 ip pim neighbor-filter 1

access-list 1 deny 10.0.0.2
```

```
interface serial 0/0
 ip pim sparse-dense-mode

interface fastethernet 0/0
 ip pim sparse-dense-mode
 ip igmp helper-address 10.0.0.1
```

## Bidir PIM

- Based only on shared tree, no STP (many to many, receivers are also senders)
- Source sends traffic unconditionaly to RP at any time (no registration process like in SM)
- Designranted 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 manualy, with BSR or Auto-RP. For the the automatic methods, a ***bidir*** keyword is required at the end (***send-rp-announce*** and ***rp-candidate***)

## SSM

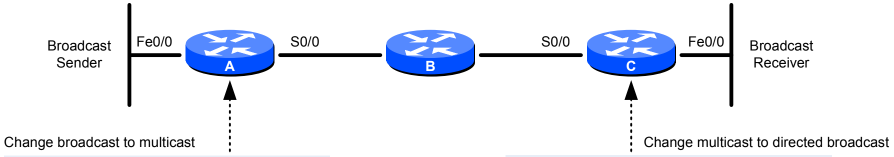
- Does not require RP (no shared trees). Only Source trees are built
- Only edge routers must support SSM, other routers only require PIM-SM
- ***ip igmp version 3***
  Requires IGMPv3 (INCLUDE/EXCLUDE messages). Hosts can decide which sources they want to join explicitly. The (*,G) joins are dropped.
- ***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

## DVMRP

- Uses IGMP v1 messages to carry routing information. Metric is a hop-count like in RIP.
- Router sends periodical reports with a list of directly connected subnets
- Routes received via DVMRP are only used for RPF, not for directing traffic toward destination
- PIM routers automaticaly discover DVMRP peers on attached interfaces
- Not fully implemented on IOS. Can be enabled only on edge routers and interfaces to peer with DVMRP-capable legacy devices
- ***ip dvmrp unicast-routing***
  Enable DVMRP routes to take precedence over unicast routes for checking RPF
- ***(IF) 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.
- ***(IF) no ip dvmrp auto-summary***
  Like in RIP, routes are automaticaly summarized
- To connect to MBone, a unidirectional tunnel can be configured (***tunnel mode dvmrp***).

## 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 automaticaly forwarded. To enable additional UDP port ***ip forward protocol <port number>*** must be added on edge routers.



Change broadcast to multicast

```
interface fastethernet 0/0
 ip multicast helper-map broadcast 224.1.2.3 100

ip forward protocol 5555

access-list 100 permit udp any any 5555
```

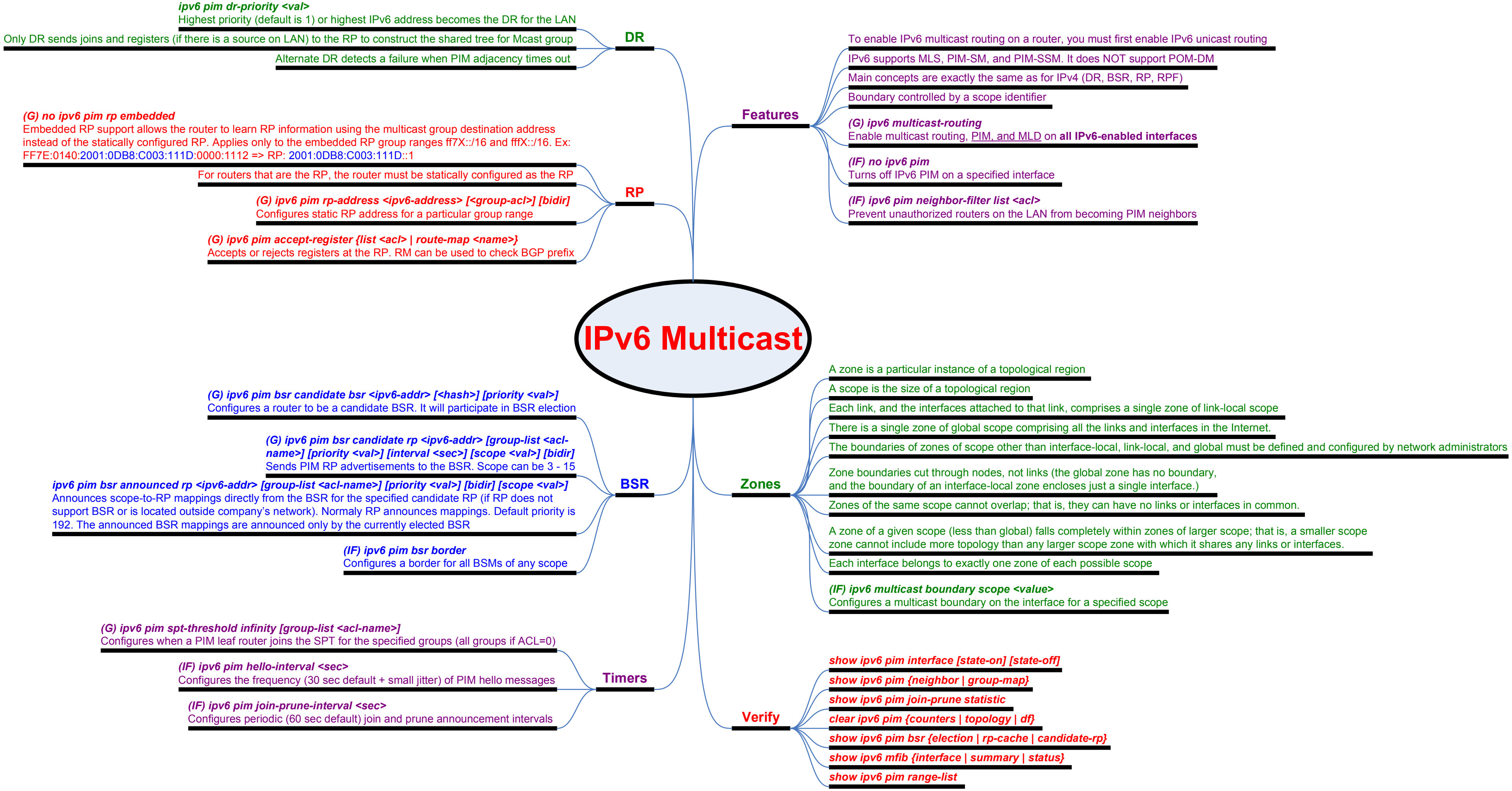
Change multicast to directed broadcast

```
interface serial 0/0
 ip multicast helper-map 224.1.2.3 10.0.0.255 100

interface fstethernet 0/0
 ip directed broadcast
 ip address 10.0.0.1 255.255.255.0

ip forward protocol 5555

access-list 100 permit udp any any 5555
```



# IPv6 Multicast

## DR

- ***ipv6 pim dr-priority <val>***
  Highest priority (default is 1) or highest IPv6 address becomes the DR for the LAN
- Only DR sends joins and registers (if there is a source on LAN) to the RP to construct the shared tree for Mcast group
- Alternate DR detects a failure when PIM adjacency times out

## RP

- ***(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 ff7X::/16 and fffX::/16. Ex: FF7E:0140:2001:0DB8:C003:111D:0000:1112 => RP: 2001:0DB8:C003:111D::1
- For routers that are the RP, the router must be statically configured as the RP
- ***(G) ipv6 pim rp-address <ipv6-address> [<group-acl>] [bidir]***
  Configures static RP address for a particular group range
- ***(G) ipv6 pim accept-register {list <acl> | route-map <name>}***
  Accepts or rejects registers at the RP. RM can be used to check BGP prefix

## BSR

- ***(G) ipv6 pim bsr candidate bsr <ipv6-addr> [<hash>] [priority <val>]***
  Configures a router to be a candidate BSR. It will participate in BSR election
- ***(G) ipv6 pim bsr candidate rp <ipv6-addr> [group-list <acl-name>] [priority <val>] [interval <sec>] [scope <val>] [bidir]***
  Sends PIM RP advertisements to the BSR. Scope can be 3 - 15
- ***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 (if RP does not support BSR or is located outside company's network). Normaly RP announces mappings. Default priority is 192. The announced BSR mappings are announced only by the currently elected BSR
- ***(IF) ipv6 pim bsr border***
  Configures a border for all BSMs of any scope

## Timers

- ***(G) ipv6 pim spt-threshold infinity [group-list <acl-name>]***
  Configures when a PIM leaf router joins 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 join-prune-interval <sec>***
  Configures periodic (60 sec default) join and prune announcement intervals

## Features

- 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)
- Boundary controlled by a scope identifier
- ***(G) ipv6 multicast-routing***
  Enable multicast routing, PIM, and MLD on **all IPv6-enabled interfaces**
- ***(IF) no ipv6 pim***
  Turns off IPv6 PIM on a specified interface
- ***(IF) ipv6 pim neighbor-filter list <acl>***
  Prevent unauthorized routers on the LAN from becoming PIM neighbors

## Zones

- 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
- 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
- Zone boundaries cut through nodes, not links (the global zone has no boundary, and the boundary of an interface-local zone encloses just a single interface.)
- Zones of the same scope cannot overlap; that is, they can have no links or interfaces in common.
- 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

## Verify

- ***show ipv6 pim interface [state-on] [state-off]***
- ***show ipv6 pim {neighbor | group-map}***
- ***show ipv6 pim join-prune statistic***
- ***clear ipv6 pim {counters | topology | df}***
- ***show ipv6 pim bsr {election | rp-cache | candidate-rp}***
- ***show ipv6 mfib {interface | summary | status}***
- ***show ipv6 pim range-list***

# MLD

## Features

- Used by IPv6 routers to discover multicast listeners on directly attached links
- MLDv1 is based on IGMPv2 for IPv4. MLDv2 is based on IGMPv3 for IPv4, and is fully backward-compatible with v1
- MLD uses ICMPv6 to carry its messages. All MLD messages are link-local with a TTL=1. Router alert option is set
- 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
  - Sending reports with the unspecified address (::) is allowed to support IPv6 multicast in the NDP
- Done
  - multicast address field is set to specific IPv6 multicast address to which the host was listening
  - If MLDv1 host sends Leave message the router must send query to ask if there are other listeners. It is 2 sec "leave latency" – last member query intervel 1 sec, query sent twice

## Timers

- ***ipv6 mld query-interval <sec>***
  Configures the frequency (125 sec default) at which the Cisco IOS software sends MLD host-query messages (only DR for LAN)
- ***ipv6 mld query-timeout <sec>***
  Configures the timeout (250 sec default) value before the router takes over as the querier for the interface
- ***ipv6 mld query-max-response-time <sec>***
  Configures the maximum (10 sec default) response time advertised in MLD queries. Defines how much time hosts have to answer an MLD query message before the router deletes their group

## Config

- ***(IF) ipv6 mld access-group <ACL-name>***
  Multicast receiver access control. State is not created for denied groups
- ***(IF) ipv6 mld join-group [<group>] [include | exclude] {<source-ip> | source-list [<acl>]}***
  Configures MLD reporting for a specified group and source. Useful for hosts not supporting MLD. Pingable
- ***(IF) ipv6 mld static-group [<group>] [include | exclude] {<source-ip> | source-list [<acl>]}***
  Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface. Non-pingable.
- ***(IF) ipv6 mld explicit-tracking <ACL-name>***
  The explicit tracking allows a router to track hosts and enables the fast leave mechanism with MLDv2 host reports. ACL defines group range for which explicit tracking can be enabled
- ***(IF) no ipv6 mld router***
  Disables MLD router-side processing on a specified interface. PIM is still enabled.

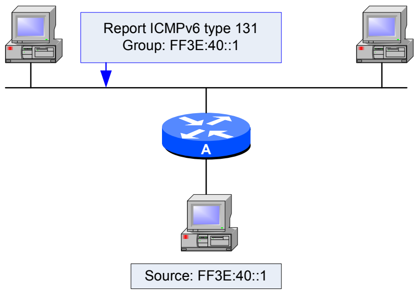
## Limiting

- Per-interface and global MLD limits operate independently. Both limits are disabled by default
- ***(G) ipv6 mld state-limit <#>***
  Limits the number of MLD states globally
- ***(IF) ipv6 mld limit <#> [except <acl>]***
  Limits the number of MLD states per-interface

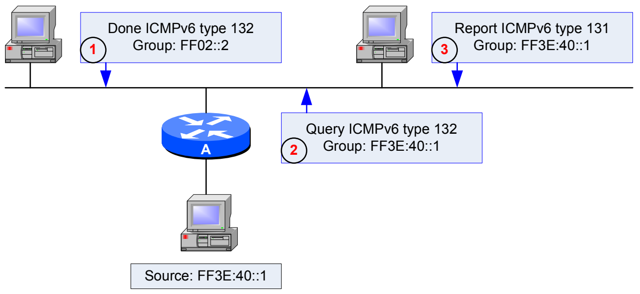
## Verify

- ***show ipv6 mld groups summary***
- ***show ipv6 mld interface [<if>]***
- ***{show | clear} ipv6 mld traffic***
- ***clear ipv6 mld counters [<if>]***

Host joining Mcast group



Host leaving Mcast group



# Classify & Mark

## IPv4

- Assured Forwarding AFxy => DSCP = 8*x + 2*y
- 8 bits TOS byte in IP header
- 3 bits IP Precedence (class selector) in TOS byte of IP header
- 6 bits DSCP in TOS byte of IP header

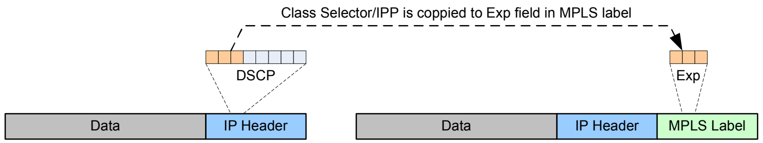
| TOS | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| IP Prec | | |
|---|---|---|
| 2 | 1 | 0 |

| DSCP | | | | | | ECN | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 4 | 3 | 2 | 1 | 0 | 1 | 0 | **DSCP** | **TOS** | **IPP** | | **PHB** |
| 1 | 1 | 1 | 0 | 0 | 0 | | | **56** | 224 | 7 | Network control | CS7 |
| 1 | 1 | 0 | 0 | 0 | 0 | | | **48** | 192 | 6 | Internetwork control | CS6 |
| 1 | 0 | 1 | 1 | 1 | 0 | | | **46** | 184 | | | EF |
| 1 | 0 | 1 | 0 | 0 | 0 | | | **40** | 160 | 5 | Critical | CS5 |
| 1 | 0 | 0 | 1 | 1 | 0 | | | **38** | 152 | | | AF43 |
| 1 | 0 | 0 | 1 | 0 | 0 | | | **36** | 144 | | | AF42 |
| 1 | 0 | 0 | 0 | 1 | 0 | | | **34** | 136 | | | AF41 |
| 1 | 0 | 0 | 0 | 0 | 0 | | | **32** | 128 | 4 | Flash override | CS4 |
| 0 | 1 | 1 | 1 | 1 | 0 | | | **30** | 120 | | | AF33 |
| 0 | 1 | 1 | 1 | 0 | 0 | | | **28** | 112 | | | AF32 |
| 0 | 1 | 1 | 0 | 1 | 0 | | | **26** | 104 | | | AF31 |
| 0 | 1 | 1 | 0 | 0 | 0 | | | **24** | 96 | 3 | Flash | CS3 |
| 0 | 1 | 0 | 1 | 1 | 0 | | | **22** | 88 | | | AF23 |
| 0 | 1 | 0 | 1 | 0 | 0 | | | **20** | 80 | | | AF22 |
| 0 | 1 | 0 | 0 | 1 | 0 | | | **18** | 72 | | | AF21 |
| 0 | 1 | 0 | 0 | 0 | 0 | | | **16** | 64 | 2 | Immediate | CS2 |
| 0 | 0 | 1 | 1 | 1 | 0 | | | **14** | 56 | | | AF13 |
| 0 | 0 | 1 | 1 | 0 | 0 | | | **12** | 48 | | | AF12 |
| 0 | 0 | 1 | 0 | 1 | 0 | | | **10** | 40 | | | AF11 |
| 0 | 0 | 1 | 0 | 0 | 0 | | | **8** | 32 | 1 | Priority | CS1 |
| 0 | 0 | 0 | 0 | 0 | 0 | | | **0** | 0 | 0 | Routine | BE |

## MPLS

- 3 bits MPLS Experimental field



## MQC

- Class-map
  - bandwidth command is not policed. If there is no congestion, class can use more bandwidth
  - Up to 4 COS or IPP vlaues can be set in one match cos/precedence statement
  - Up to 8 DSCP vlaues can be set in one match dscp statement
- FIFO is required on physical interface. MQC is not compatible with other per-interface queues
- ***class <nameA>***
  ***match not class <nameB>***
- ***class-map match-any <name>***
  If ANY match statement within a class is matched, the class is executed
- ***class-map match-all <name>***
  The class is executed only if ALL match statement within a class are matched
- ***match ip prec 1 2 3***
  Any of specified IP Precedences needs to be matched (logical OR).

## Pre-classification

- TOS field is copied from original field
- ***qos pre-classify***
  - ***crypto-map*** - IPSec
  - ***interface tunnel*** - GRE
  - ***interface virtual-template*** – L2TP, L2F

## NBAR

- 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 protocol http mime image/jpeg***
  This would match jpeg,jpg,jpe,jfif,pjpeg, and pjp types
- ***show ip nbar protocol-discovery***
- ***ip nbar port-map <protocol-name> [tcp | udp] <port-number>***
  Use a different port number than the well-known port
- ***ip nbar pdlm <pdlm-name>***
  Extends the list of protocols recognized by NBAR by adding additional PDLMs

## L2

- FR DE
  - MQC
    - ***(class) set fr-de***
    - Applies to all packet-switching paths including CEF
  - Legacy
    - ***frame-relay de-group <#> <dlci>***
    - ***frame-relay de-list <#> protocol ip ...***
    - Applies only to process-switched packets
- Ethernet
  - 3 bits COS in 802.1/ISL frames. Possible only on trunk links, where 802.1q tag and ISL encapsulation exist
  - If policy-map is applied, all other QOS features are disabled on interface
    except default COS marking, which is used for ***trust cos*** option within classes
  - 3560
    - ***(G) mls qos cos policy-map***
      Must be enabled to set COS in policy-maps
    - Treats IPv6 as IP traffic
    - Class-default catches all IP and non-IP
  - 3550
    - Treats IPv6 as non-IP traffic
    - Class-default is unpredictible, so additional class should be created to catch all IP and non-IP traffic (MAC ACL)

# CBWFQ

## Concept

- Max 64 queues/classes (63 + class-default)
- WRED can be enabled on all queues (but not LLQ)
- FIFO within each queue except class-default (FIFO or WFQ)
  - ***queue-limit <#>***
    Max packets per class (threshold for tail drop). Default is 256. Only power of 2 is accepted. It cannot be configured with WRED.
  - ***fair-queue [<# of dynamic conv>]***
    In class-default only

## LLQ

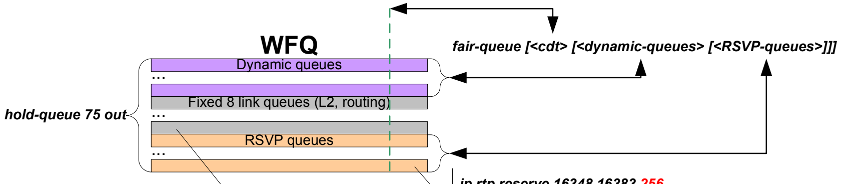
- ***priority {<bw> | percent <%>} [<burst>]***
  burst by default 20% of configured priority BW. Burst is actualy max packet size
- Policies traffic up to defined priority BW
- BW + PQ is still limited to 75% of intf BW
- Unlike bandwidth, priority can use percent and remaining-percent in the same policy at the same time

## WFQ

- 4096 queues. Automatic classification based on flows. eight hidden queues (very low weight) for overhead traffic generated by the router
- To provide fairness, WFQ gives each flow an equal amount of bandwidth
- Queues with lower volume and higher IP precedence get more service. If one flow is marked with Prec 0 and the other with Prec 1, the latter one will get twice the bandwidth of the first one.
- The WFQ scheduler takes the packet with the lowest sequence number (SN) among all the queues, and moves it to the Hardware Queue
- WFQ scheduler considers packet length and precedence when calculating SN. Calculation results in a higher number for larger packets
  *SN = Previous_SN + (weight * new_packet_length)*
  *Weight = [32,384 / (IP_Precedence + 1)]*
- L2 header is added to calculations

```
show interface serial0/0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
    Conversations  0/0/256 (active/max active/max total)
    Reserved Conversations 0/0 (allocated/max allocated)
    Available Bandwidth 1158 kilobits/sec
```

- ***(IF) hold-queue <len> out*** - absolute number of packets in whole WFQ per intf
- ***(IF) fair-queue [<cdt> [<dynamic-queues> [<RSVP-queues>]]]***
- CDT – Congestion avoidance scheme available in WFQ. When CDT threshold is reached WFQ drops packet from a flow queue with max virtual scheduling time.
- Once traffic is emptied from one flow queue, the flow queue is removd, even if TCP session between two hosts is still up
- Modified tail drop
  - If a packet needs to be placed into a queue, and that queue's CDT (1-4096) has been reached, the packet may be thrown away
  - 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 enqueued in a different queue, however, WFQ instead discards the packet with the larger SN



WFQ
Dynamic queues
...
Fixed 8 link queues (L2, routing)
...
RSVP queues
...
hold-queue 75 out
fair-queue [<cdt> [<dynamic-queues> [<RSVP-queues>]]]

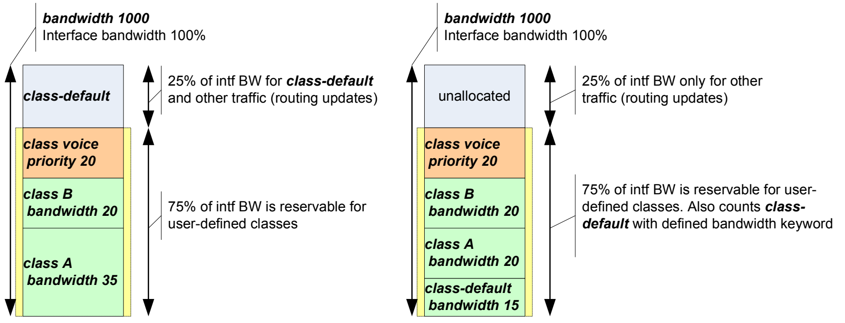
***ip rtp reserve 16348 16383 256***
One RSVP queue is reserved for RTP traffic. This queue gets weight 128 and is policed up to 256k (exceeding traffic gets weight 32384). Voice still may compete with other flows

***ip rtp priority 16348 16383 256***
This queue gets weight 0 and is policed up to 256k. Also, only even UDP ports are considered. Voice always gets priority. This queue sits just right after 8 link queues

## BW

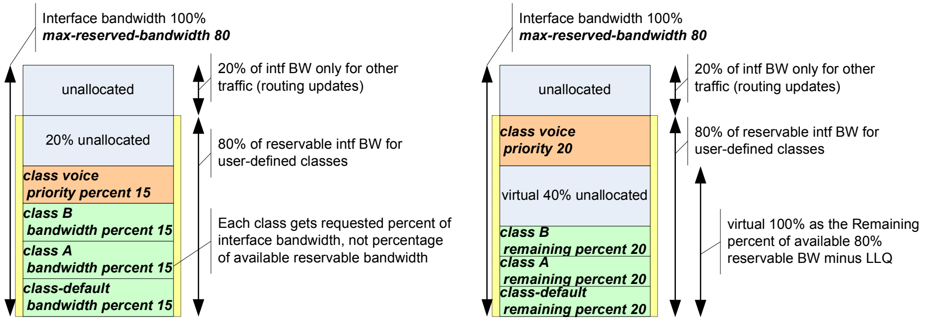
- If one queue does not currently allocate BW its resources are distributed for other queues proportionaly to configured bandwidth
- Only one variation of BW can be used (static or percentage)
- Percent
  - ***bandwidth percent <%>*** - Always % of literal interface BW
  - ***bandwidth remaining-percent <%>***
    % of reservable BW (int-bw * max-res) minus already reserved BW.
- Max reservable BW for non-class-default queues – 75%
  - ***(IF) max-reserved-bandwidth <%>***
  - If class-default has bandwidth defined it is also calculated as reservable

Static bandwidth configuration with BW assigned to class-default and not



bandwidth 1000
Interface bandwidth 100%
class-default
class voice
priority 20
class B
bandwidth 20
class A
bandwidth 35
25% of intf BW for class-default and other traffic (routing updates)
75% of intf BW is reservable for user-defined classes
bandwidth 1000
Interface bandwidth 100%
unallocated
class voice
priority 20
class B
bandwidth 20
class A
bandwidth 20
class-default
bandwidth 15
25% of intf BW only for other traffic (routing updates)
75% of intf BW is reservable for user-defined classes. Also counts class-default with defined bandwidth keyword

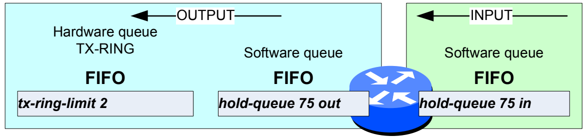
percentage bandwidth configuration with bandwidth percent and remaining percent



Interface bandwidth 100%
max-reserved-bandwidth 80
unallocated
20% unallocated
class voice
priority percent 15
class B
bandwidth percent 15
class A
bandwidth percent 15
class-default
bandwidth percent 15
20% of intf BW only for other traffic (routing updates)
80% of reservable intf BW for user-defined classes
Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth
Interface bandwidth 100%
max-reserved-bandwidth 80
unallocated
class voice
priority 20
virtual 40% unallocated
class B
remaining percent 20
class A
remaining percent 20
class-default
remaining percent 20
20% of intf BW only for other traffic (routing updates)
80% of reservable intf BW for user-defined classes
virtual 100% as the Remaining percent of available 80% reservable BW minus LLQ

# FIFO

- 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.
- ***(IF) tx-ring-limit <#packets>***
- The smaller the value, the less impact the TX Queue has on the effects of the queuing method.
- ***tx_limited=0(16)***
  TX Ring is here 16 packets. Zero means that the queue size is not limited due to queuing tool enabled on the intf. IOS shrinks tx-queue if software Q is applied on intf to give more control to SW Q

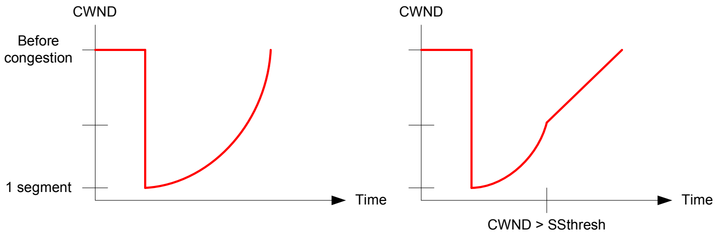


- Input interface queue is always FIFO (default 75 packets)
- ***(IF) hold-queue <#> {in | out}***
- ***no fair-queue***

# WRED

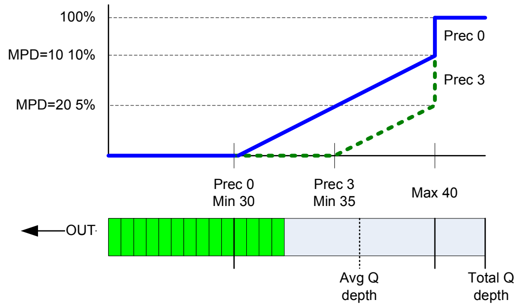
## TCP behaviour

- 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) - not communicated. 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 receiving an ack. CWND is designed to let the sender react to network congestion by slowing down its sending rate
- 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 SSTHRESH is set to 50% of CWND value before lost segment
  4) Slow start governs how fast CWND grows until it reaches value of SSTHRESH.
  5) After CWND > SSTHRESH 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



**Slow-start only**

**Slow-start and congestion avoidance**



## MPD mark probability denominator

- 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
- ***random-detect exponential-weighting-constant <val>***
- RED decides whether to discard packets by comparing the average queue depth to two thresholds, called the minimum threshold and maximum threshold.

## ECN

- 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, telling it to slow down. TCP sender reduces CWND by half.
- **5)** TCP sender sets Congestion Window Reduced (CWR) flag in next segment to inform receiver it slowed down
- ***random-detect dscp-based***
  ***random-detect ecn***

## Configuration

### Legacy

- Can be configured only on main interfaces. Sets FIFO on interface
- ***(IF) random-detect*** – enable RED
- ***(IF) random-detect {dscp-based | prec-based}***
- ***(IF) random-detect {dscp <dsc> | precedence <prec>} <min> <max> <mpd>***
- ***(IF) random-detect exponential-weighting-constant <val>***

### Flow-based

- ***random-detect flow***
- ***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

### MQC

- ***random-detect***
- ***random-detect {dscp <dsc> | precedence <prec>} <min> <max> <mpd>***

# Shaping

## Features

- Traffic metering is based on token bucket concept
- Shaping does not count TCP/IP headers and works only in outbound direction
- FRTS applies FIFO into physical interface (WFQ is disabled, although nested CBWFQ can be used)
- CBWFQ cannot be applied to FR subinterfaces, but if applied to physical interface, ***match fr-dlci*** can be used
- ISP usualy polices input rate, and the customer usualy 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)
- If Be is used, overflowed tokens from Bc bucket are put into Be bucket. Bc + Be bytes can be sent during one Tc
- During congestion adaptive shaping can drop traffic to minimum rate defined by MinCIR (50% of CIR by default)
- ***(IF) frame-relay broadcast-queue <size> <Bps> <packet-rate>***
  Frame-relay broadcast queue is an interface-level priority queue for L3 packets which need to be replicated to all VCs on L2 level (routing updates). Default is 64 packets, byte-rate: 256000 bps at 36 packets per second
-
```
Target  Byte   Sustain  Excess  Interval  Increment  Adaptive
 Rate   Limit   Rate     Bits      ms       Bytes     Active
 CIR    Bc+Be   Bc       Be        Tc       Bc          -
        -----                               --
          8                                 8
```
- If frame-relay traffic shaping is enabled, all VCs are affected by default configuration. How VC is affected depends on where map-class is applied. MQC has to be used if only one VC is to be using shaping

```
interface serial0/0
 encapsulation frame-relay
 frame-relay traffic-chaping
 frame-relay class C2

interface serial0/0.1 point-to-point
 frame-relay class C1
 frame-relay interface-dlci 101

interface serial0/0.2 point-to-point
 frame-relay interface-dlci 102

interface serial0/0.3 multipoint
 frame-relay class C2
 frame-relay interface-dlci 103
 frame-relay interface-dlci 104
  class C3
```

Default CIR=56kbps, Bc=7000bits, Tc=125ms

## Class-based

- ***shape average <CIR bps> [<Bc>] [<Be>]***
  Be is available if there were periods of inactivity and tokens were collected. Tc = Bc/CIR. If Be is omited it is the same as Bc, so it should be „0" if it's not used (unlike in FRTS where Be is 0 by default)
- ***class class-default***
  ***shape average <CIR bps> [<Bc>] [<Be>]***
  ***service-policy <name>***
  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
- ***shape peak <mean rate> [<Bc>] [<Be>]***
  Refils Bc + Be every Tc. PIR = CIR*(1 + Be/Bc). If Be is omited it is the same as Bc, so PIR = 2*CIR
- ***shape max-buffers <#>***
  Max queue length for the default FIFO shaping queue
- ***shape adaptive <minCIR>***

## Legacy generic FR shaping

- ***(IF) traffic-shape rate <cir> <Bc> <Be> <QueueLimit>***
  Frame-relay encapsulation has to be configured on the interface
- ***(IF) traffic-shape adaptive <minCIR>***
  Adaptive keyword becomes available only if FR encapsulation is used. It reacts to BECN
- ***(IF) traffic-shape fecn-adapt***
  For unidirectional traffic BECN cannot be sent, so Q922 test frame can be sent by routers as reaction for FECN (FECN reflection)
- All PVCs are shaped. GTS for FR is not true per-VC FR traffic shaping. Can be applied to physical interface or subinterfaces

## Legacy FR shaping

- Be is 0 by default. Minimum possible Tc is 10ms for FR (set Bc to CIR/100 value)
- ***frame-relay traffic-shaping***
  Required on physical interface, regardless of where map-class is applied (actualy enables FRTS)
- Basic configuration
  - ***map-class frame-relay <name>***
    ***frame-relay cir <cir>***
    ***frame-relay bc <Bc>***
    ***frame-relay be <Be>***
  - ***(map-class) frame-relay holdq <#>***
    Number of buffers dedicated for traffic shaping
- Adaptive shaping
  - ***(map-class) frame-relay mincir <minCIR>***
    Define minCIR to which shaping rate drops after adaptive condition is met (CIR/2 default)
  - ***(map-class) frame-relay adaptive-shaping becn*** – react to BECN
  - ***(map-class) frame-relay adaptive-shaping interface-congestion [<packets>]***
    React to interface congestion (FIFO queue). If number of packets (default is 0) exceed defined value, adaptive condition is met.
  - Each time BECN or Foresight is received, rate drops by 25%
  - Dropping occurs until MinCIR (MinCIR should be the same as „CIR" defined by telco)
  - Traffic rate grows [(Bc + Be) / 16] after consecutive 16 Tc without BECN until CIR
- FRTS + CBWFQ/LLQ
  - All classes within CBWFQ are processed by the scheduler, and then all outgoing packets are shaped
  - Dual-FIFO on physical interface is used to serve voice packets first (everything from priority queue inside CBWFQ is placed into FIFO priority queue during shaping)
  - If service-policy is used within ***map-class frame-relay*** when FRTS is used, then minCIR is used as an available BW for CBWFQ
  - ***map-class frame-relay <name>***
    ***frame-relay cir <cir>***
    ***service-policy output <cbwfq+llq policy>***

## Other Legacy features

- Per-VC WFQ
  - The interface queue is FIFO, but each VC can be configured with own WFQ
  - ***map-class frame-relay <name>***
    ***frame-relay fair-queue <cdt> <flows> <rsvp flows> <max buffers>***
- Per-VC PQ
  - ***(map-class) frame-relay priority-group <#>***
  - Broadcast traffic like RIP is automaticaly dequeued first (doesn't have to be assigned to any priority queue), as it is served by the internal FR broadcast queue
- Per-VC CQ
  - ***(map-class) frame-relay custom-queue-list <#>***
- Per-VC RTP Priority
  - ***(map-class) frame-relay ip rtp priority <first port> <range> <Bps>***
  - RTP priority is activated only if FRF.12 is configured. Defined speed in bps is policed, so other packets are not starved.
- Priority to DLCI mapping
  - ***interface serial0/0***
    ***priority-group <PQ#>***
    ***frame-relay priority-group-dlci <PQ#> <hi> <med> <normal> <low>*** ! DLCIs
- PIPQ
  - When using FR PIPQ (PVC Interface Priority Queueing), configure the network so that different types of traffic are transported on separate PVCs
  - ***(map-class) frame-relay interface-queue priority {high | medium | normal | low}***
    Define different map-classes for different VCs and assign to particular PQ queue
  - ***(IF) frame-relay interface-queue priority [<high> <med> <normal> <low>]***
    Enable PIPQ on physical interface and define limits for each PQ queue
- No Bc/Be/Tc tuning
  - ***(map-class) frame-relay traffic-rate <avg> <peak>***
    Avg is simply CIR. Peak rate is CIR + Be/Tc = CIR (1 + Be/Bc) = CIR + EIR
  - ***frame-relay traffic-rate 64000 96000***
    CIR is set to 64000 bps, Be value is 96000 - 64000 = 32000 bits

## Generic (GTS)

- ***(IF) traffic-shape {rate | group <acl>} <cir> <Bc> <Be> <QueueLimit>***
  QueueLimit sets max WFQ buffer size. WFQ is the internal queueing mechanism for GTS
- Many entries can exist on one physical interface.
- Works on all media types and encapsulations

# Policing

## Concept

Policing counts TCP/IP headers

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:
*[ (Current_packet_arrival_time – Previous_packet_arrival_time) * Police_rate ] / 8*

Default Values equivalent to sending bytes that would be transmited in 1/4 sec at defined policing rate

Multiaction

```
police <cir> <pir>
 conform-action ...
 exceed-action ---
 vialate-action set-dscp-transmit 0
 violate-action set-frde-transmit
```

For outbound policing MAC address cannot be matched with ***match source-address mac <mac>***. You can use ***match access-group <mac acl>***

## CAR

CAR can be used as policing tool, as well as multiaction marking tool (admission control)

***rate-limit {input | output} access-group <acl> <bps> <burst normal> <burst max> conform-action ... exceed-action ... violate-action ...***

To **not** to use max burst set it to the same value as burst normal, not zero

Burst should be 1/8 of speed. (Speed/8)/8 (/8 means 125 ms) as Burst is in Bytes. Bc = (CIR/8)*(Tc/1000)

Configuration statement evaluated sequentialy if ***continue*** is an action. Different rates for different IP Prec.

Sliding „averaging time interval”. New packet is confrming is already preocessed packets during that window plus current packet size is less than or equal to Bc

Tc is a constant value of 1/8000 sec. that's why values are defined in rates of 8k

L2 header is taken into consideration when calculating bandwidth.

### ACL

Each ACL can contain only one line

***rate-limit {input | output} access-group rate-limit <acl> ...***

***access-list rate-limit <#> <mac-address>***

***access-list rate-limit <#> <IP Prec hex mask>***

***TOS byte: 0001 0110 => 0x16***

## Nested policers

Up to 3 nesting policers. Upper-level policers are applied first. Packets which are not to be dropped are passed to next policer.

```
policy-map OUT
 class OUT
  police rate percent 50
  service-policy IN
```

50% of interface bandwidth

```
 policy-map IN
  class IN
   police rate percent 50
```

50% of outer policy-map

## Single-rate Two-color

One bucket, Conform, Exceed, CIR

Policing does not use  Tc but tokens. Number of tokens available is counted as:
((Current Packet Arrival Time – Previous Packet Arrival Time) * Police Rate) / 8

Tokens are replenished at policing rate (CIR)

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

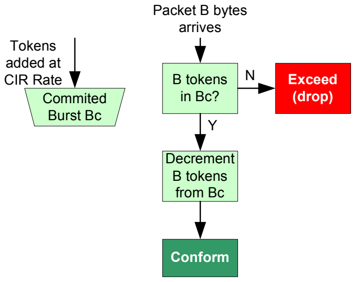
Number of bits in packet is compared to number of available tokens in a bucket. Packet is either transmited or dropped.

Default for single-bucket Bc = CIR/32 or 1500, whichever is larger, Be = 0

Default for dual-bucket: Bc = CIR/32, Be = Bc

***police 32000 1000 conform-action ...***

32000 bits / 8  = 4000 bytes per sec
4000 bytes / 1000 = 4 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)



## Single-rate Three-color

Two buckets; Three actions: Conform, Exceed, Violate

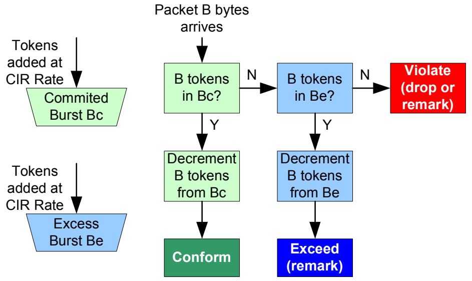
Be bucket allows bursts until Be empties

If you define Be but not violate action then Be is ignored (becomes single-rate two-color)

```
police 32000 1000 2000
 conform-action set-prec-transmit 1
 exceed-action set-dscp-transmit 0
 violate-action drop
```

CIR – how fast tokens are replenished within 1 sec

Bc and Be are not cumulative



## Two-rate Three-color

Two buckets; Three actions: Conform, Exceed, Violate; Two rates: CIR, PIR

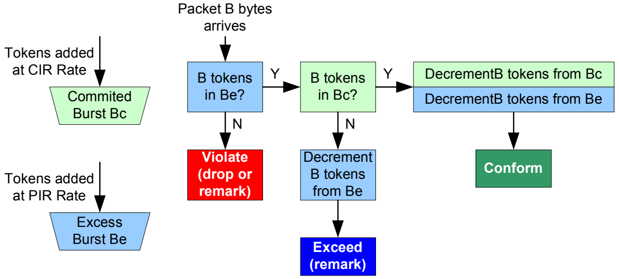
Be is filled twice faster that Bc. If Bc (CIR) = 128, then Be (PIR) = 256k. During conform action tokets are taken from both buckets

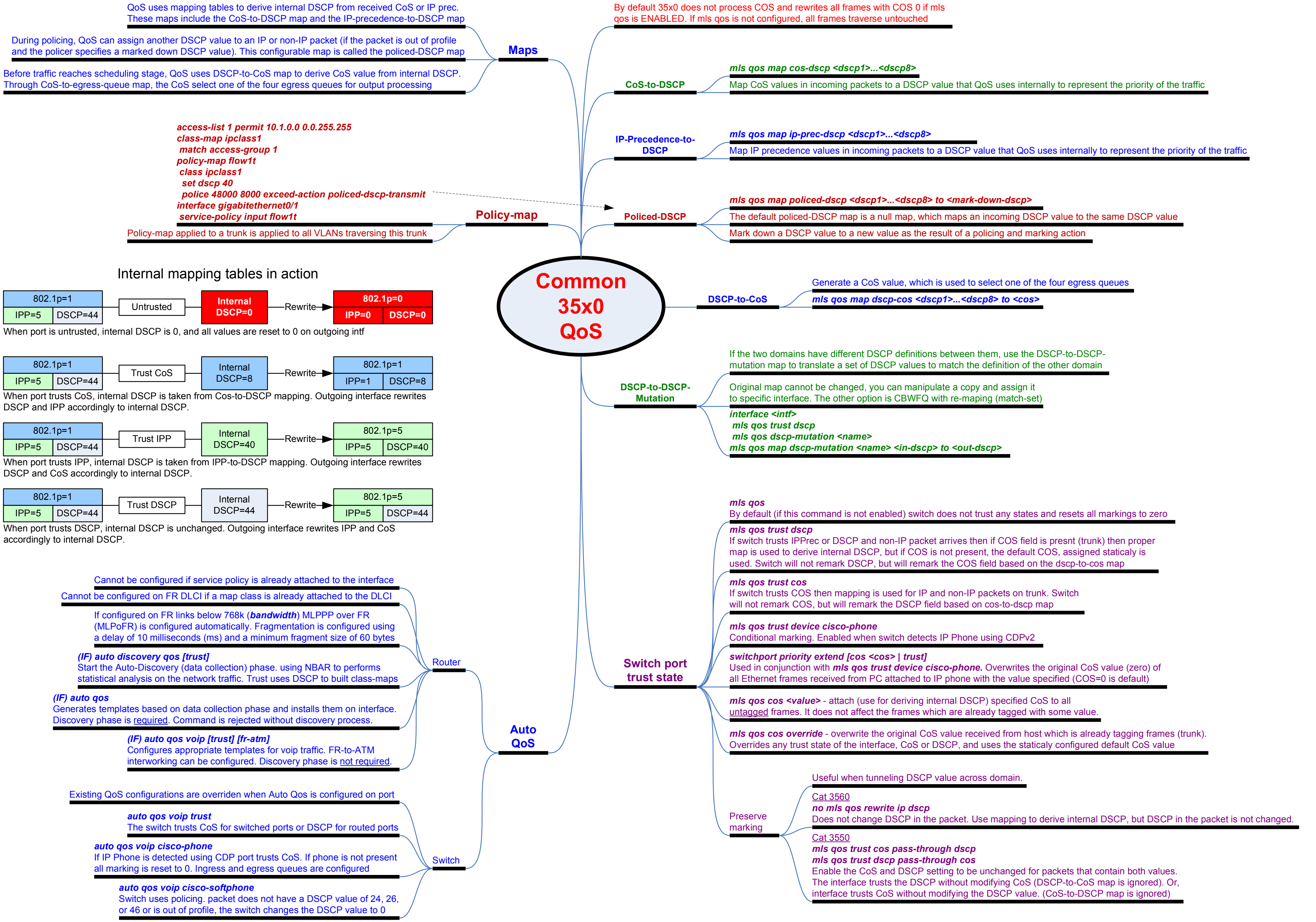
***police cir <cir> [bc <Bc>] pir <pir> [be <Be>] conform-action ...***

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

This is actualy the same as single rate two color in effect, but in addition you can collect statistics from interface to see what is the excess (business usage)

The same effect:
***police 48000***
***police cir 32000 pir 48000***





# Common 35x0 QoS

## Maps

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 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

## Policy-map

```
access-list 1 permit 10.1.0.0 0.0.255.255
class-map ipclass1
 match access-group 1
policy-map flow1t
 class ipclass1
  set dscp 40
  police 48000 8000 exceed-action policed-dscp-transmit
interface gigabitethernet0/1
 service-policy input flow1t
```

Policy-map applied to a trunk is applied to all VLANs traversing this trunk

## Internal mapping tables in action

802.1p=1 | IPP=5 | DSCP=44 — Untrusted — Internal DSCP=0 — Rewrite → 802.1p=0 | IPP=0 | DSCP=0

When port is untrusted, internal DSCP is 0, and all values are reset to 0 on outgoing intf

802.1p=1 | IPP=5 | DSCP=44 — Trust CoS — Internal DSCP=8 — Rewrite → 802.1p=1 | IPP=1 | DSCP=8

When port trusts CoS, internal DSCP is taken from Cos-to-DSCP mapping. Outgoing interface rewrites DSCP and IPP accordingly to internal DSCP.

802.1p=1 | IPP=5 | DSCP=44 — Trust IPP — Internal DSCP=40 — Rewrite → 802.1p=5 | IPP=5 | DSCP=40

When port trusts IPP, internal DSCP is taken from IPP-to-DSCP mapping. Outgoing interface rewrites DSCP and CoS accordingly to internal DSCP.

802.1p=1 | IPP=5 | DSCP=44 — Trust DSCP — Internal DSCP=44 — Rewrite → 802.1p=5 | IPP=5 | DSCP=44

When port trusts DSCP, internal DSCP is unchanged. Outgoing interface rewrites IPP and CoS accordingly to internal DSCP.

## Auto QoS

### Router

Cannot be configured if service policy is already attached to the interface

Cannot be configured on FR DLCI if a map class is already attached to the DLCI

If configured on FR links below 768k (***bandwidth***) MLPPP over FR (MLPoFR) 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 performs statistical analysis on the network traffic. Trust uses DSCP to built 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.

***(IF) auto qos voip [trust] [fr-atm]***
Configures appropriate templates for voip traffic. FR-to-ATM interworking can be configured. Discovery phase is not required.

### Switch

Existing QoS configurations are overriden 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 all marking is reset to 0. Ingress and egress queues are configured

***auto qos voip cisco-softphone***
Switch uses policing. packet does not have a DSCP value of 24, 26, or 46 or is out of profile, the switch changes the DSCP value to 0

By default 35x0 does not process COS and rewrites all frames with COS 0 if mls qos is ENABLED. If mls qos is not configured, all frames traverse untouched

## CoS-to-DSCP

***mls qos map cos-dscp <dscp1>...<dscp8>***

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

## IP-Precedence-to-DSCP

***mls qos map ip-prec-dscp <dscp1>...<dscp8>***

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

## Policed-DSCP

***mls qos map policed-dscp <dscp1>...<dscp8> to <mark-down-dscp>***

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

Mark down a DSCP value to a new value as the result of a policing and marking action

## DSCP-to-CoS

Generate a CoS value, which is used to select one of the four egress queues

***mls qos map dscp-cos <dscp1>...<dscp8> to <cos>***

## DSCP-to-DSCP-Mutation

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

Original map cannot be changed, you can manipulate a copy and assign it to specific interface. The other option is CBWFQ with re-maping (match-set)

```
interface <intf>
 mls qos trust dscp
 mls qos dscp-mutation <name>
mls qos map dscp-mutation <name> <in-dscp> to <out-dscp>
```

## Switch port trust state

***mls qos***
By default (if this command is not enabled) switch does not trust any states and resets all markings to zero

***mls qos trust dscp***
If switch trusts IPPrec or DSCP and non-IP packet arrives then if COS field is presnt (trunk) then proper map is used to derive internal DSCP, but if COS is not present, the default COS, assigned staticaly is used. Switch will not remark DSCP, but will remark the COS field based on the dscp-to-cos map

***mls 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 cos-to-dscp map

***mls qos trust device cisco-phone***
Conditional marking. Enabled when switch detects IP Phone using CDPv2

***switchport priority extend [cos <cos> | trust]***
Used in conjunction with ***mls qos trust device cisco-phone.*** Overwrites the original CoS value (zero) of all Ethernet frames received from PC attached to IP phone with the value specified (COS=0 is default)

***mls qos cos <value>*** - attach (use for deriving internal DSCP) specified CoS to all untagged frames. It does not affect the frames which are already tagged with some value.

***mls qos cos override*** - overwrite the original CoS value received from host which is already tagging frames (trunk). Overrides any trust state of the interface, CoS or DSCP, and uses the staticaly configured default CoS value

### Preserve marking

Useful when tunneling DSCP value across domain.

Cat 3560
***no mls qos rewrite ip dscp***
Does not change DSCP in the packet. Use mapping to derive internal DSCP, but DSCP in the packet is not changed.

Cat 3550
***mls qos trust cos pass-through dscp***
***mls qos trust dscp pass-through cos***
Enable the CoS and DSCP setting to be unchanged for packets that contain both values. The interface trusts the DSCP without modifying CoS (DSCP-to-CoS map is ignored). Or, interface trusts CoS without modifying the DSCP value. (CoS-to-DSCP map is ignored)

# 3560 QoS

## VLAN based

*intervace VLAN <id>*
*service-policy in <policy-map>*

*(IF) mls qos vlan-based*
All ports assigned to the VLAN will inherit QoS from appropriate SVI

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

## Ingress Queue

Scheduler - Shared Round Robin; Sharing is the only supported mode.

Two global FIFO queues for all interfaces, one can be priority.

**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 (after threshold 1 is reached). Threshold 3 is always 100% (non-modifiable)
***mls qos srr-queue input threshold <Q1/2> <t1 %> <t2 %>***

**2. Assign COS/DSCP to thresholds**
Third threshold is 100% an cannot be changed, but COS/DSCP can be assigned to it
***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. Define memory buffers**
Ratio which divides the ingress buffers between the two queues. The buffer and the bandwidth allocation control how much data can be buffered before packets are dropped
***mls qos srr-queue input buffers <Q1%> <Q2%>***

**4. Define bandwidth**
How much of available bandwidth is allocated between ingress queues. Ratio of weights is the ratio of the frequency in which SRR scheduler sends packets from each queue
***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

4 per-interface queues with classification based on COS (Q1 can be PQ)

Two templates (queue-set). Set 1 is a default applied to all interfaces. Set 2 can be manipulated and assigned to selected interfaces. If Set 1 is manipulated, all interfaces are affected

Shaped

***(IF) srr-queue bandwidth shape <w1> <w2> <w3> <w4>***
Rate-limits queue up to queue bandwidth, even if other queues are empty. Weights are in inverse ration; 8 means 1/8 of BW

***(IF) srr-queue bandwidth shape 8 0 0 0***
Q1 is policed up to 1/8 of BW. Other queues are not policed at all. Remaining BW from those queues is shaped according to weights defined in share command. Used to define priority queue (***priority-queue out*** must be used on interface)

Shared

***(IF) srr-queue bandwidth share <w1> <w2> <w3> <w4>***
If some queues are empty, its resources will be spread across other queues proportionaly. PQ can consume whole BW. Queues are shaped

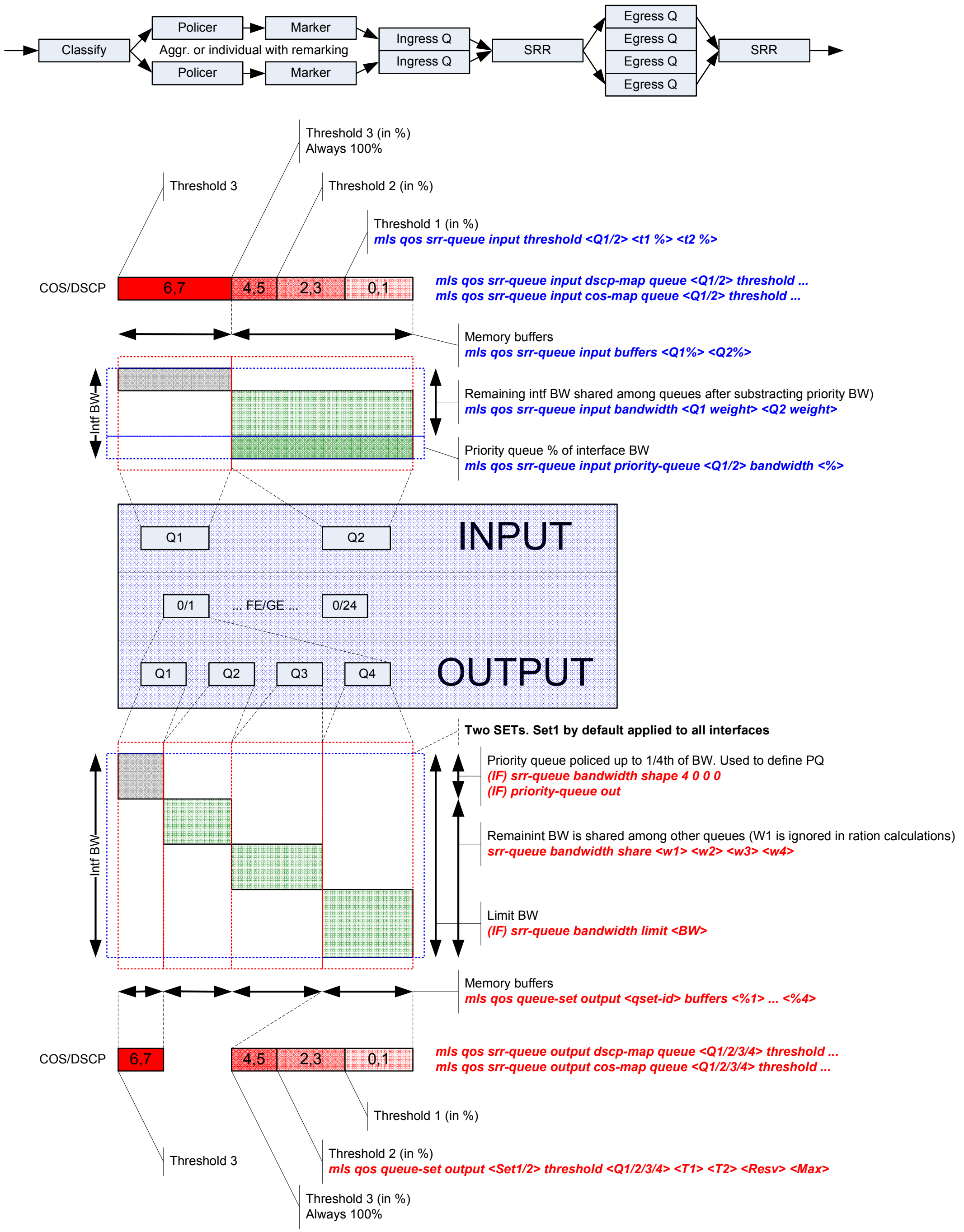
**1. Define thersholds**
Configure the WTD thresholds, guarantee the availability of buffers, and configure the maximum memory allocation for the queue-set. If one port has empty resources (nothing is plugged in) they can be used. Reserved – what each port gets on start; Max – if needed, up to this %-age can be assigned
***mls qos queue-set output <Set1/2> threshold <Q1/2/3/4> <T1> <T2> <Resv> <Max>***

**2. Assign COS/DSCP to thresholds**
Third threshold is 100% an cannot be changed, but COS/DSCP can be assigned to it
***mls qos srr-queue output dscp-map queue <Q1/2/3/4> threshold <T1/2/3> <dscp1-8>***
***mls qos srr-queue output cos-map queue <Q1/2/3/4> threshold <T1/2/3> <cos1-8>***

**3. Allocate memory buffers**
All buffers must sum up with 100%
***mls qos queue-set output <qset-id> buffers <%1> ... <%4>***

**4. Limit bandwidth**
Configurable 10-90% of physical BW on 6Mb basis. If you define 10, the limit will be 6-12Mb
***srr-queue bandwidth limit <BW>***

***(IF) queue-set {1 | 2}***
Assign queue set to an interface



Classify → Policer → Marker; Policer → Marker — Aggr. or individual with remarking → Ingress Q, Ingress Q → SRR → Egress Q, Egress Q, Egress Q, Egress Q → SRR

Threshold 3 (in %)
Always 100%

Threshold 3

Threshold 2 (in %)

Threshold 1 (in %)
***mls qos srr-queue input threshold <Q1/2> <t1 %> <t2 %>***

COS/DSCP 6,7 | 4,5 | 2,3 | 0,1

***mls qos srr-queue input dscp-map queue <Q1/2> threshold ...***
***mls qos srr-queue input cos-map queue <Q1/2> threshold ...***

Memory buffers
***mls qos srr-queue input buffers <Q1%> <Q2%>***

Intf BW

Remaining intf BW shared among queues after substracting priority BW)
***mls qos srr-queue input bandwidth <Q1 weight> <Q2 weight>***

Priority queue % of interface BW
***mls qos srr-queue input priority-queue <Q1/2> bandwidth <%>***

INPUT: Q1, Q2

0/1 ... FE/GE ... 0/24

OUTPUT: Q1, Q2, Q3, Q4

**Two SETs. Set1 by default applied to all interfaces**

Intf BW

Priority queue policed up to 1/4th of BW. Used to define PQ
***(IF) srr-queue bandwidth shape 4 0 0 0***
***(IF) priority-queue out***

Remainint BW is shared among other queues (W1 is ignored in ration calculations)
***srr-queue bandwidth share <w1> <w2> <w3> <w4>***

Limit BW
***(IF) srr-queue bandwidth limit <BW>***

Memory buffers
***mls qos queue-set output <qset-id> buffers <%1> ... <%4>***

COS/DSCP 6,7 | 4,5 | 2,3 | 0,1

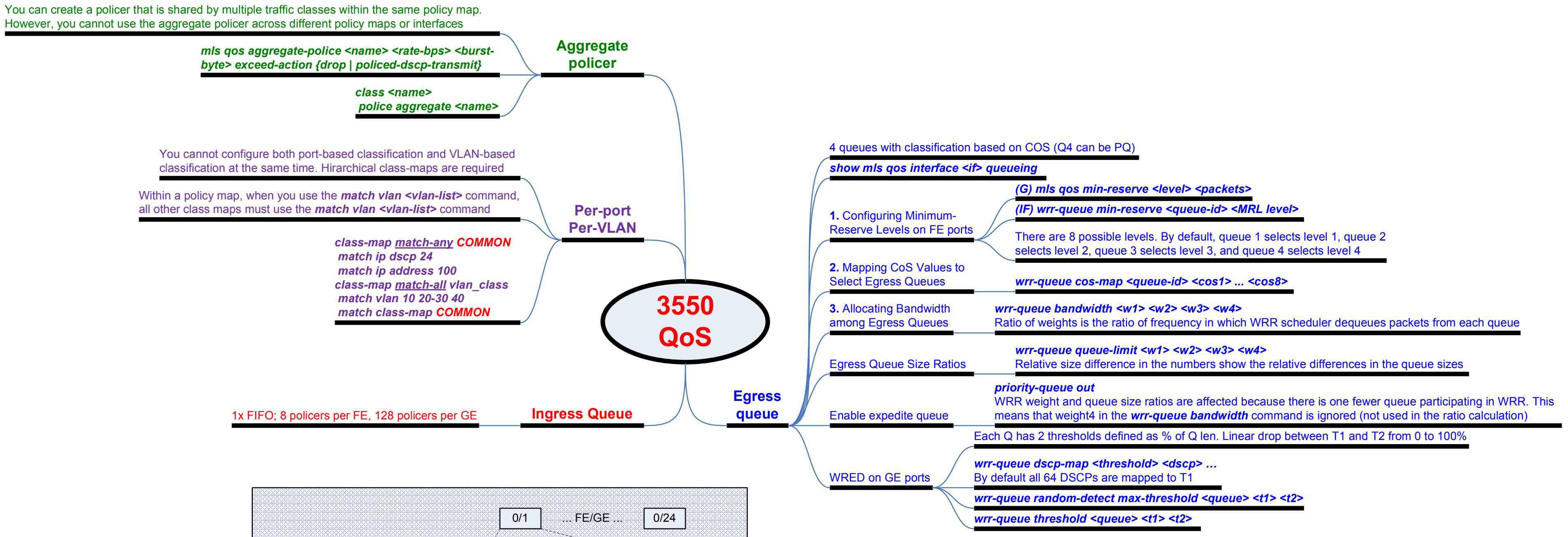
***mls qos srr-queue output dscp-map queue <Q1/2/3/4> threshold ...***
***mls qos srr-queue output cos-map queue <Q1/2/3/4> threshold ...***

Threshold 1 (in %)

Threshold 3

Threshold 2 (in %)
***mls qos queue-set output <Set1/2> threshold <Q1/2/3/4> <T1> <T2> <Resv> <Max>***

Threshold 3 (in %)
Always 100%



# 3550 QoS

## Aggregate policer

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>*
*police aggregate <name>*

## Per-port Per-VLAN

You cannot configure both port-based classification and VLAN-based classification at the same time. Hirarchical 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

```
class-map match-any COMMON
 match ip dscp 24
 match ip address 100
class-map match-all vlan_class
 match vlan 10 20-30 40
 match class-map COMMON
```

## Ingress Queue

1x FIFO; 8 policers per FE, 128 policers per GE

## Egress queue

4 queues with classification based on COS (Q4 can be PQ)

***show mls qos interface <if> queueing***

**1.** Configuring Minimum-Reserve Levels on FE ports
- ***(G) mls qos min-reserve <level> <packets>***
- ***(IF) wrr-queue min-reserve <queue-id> <MRL level>***
- 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

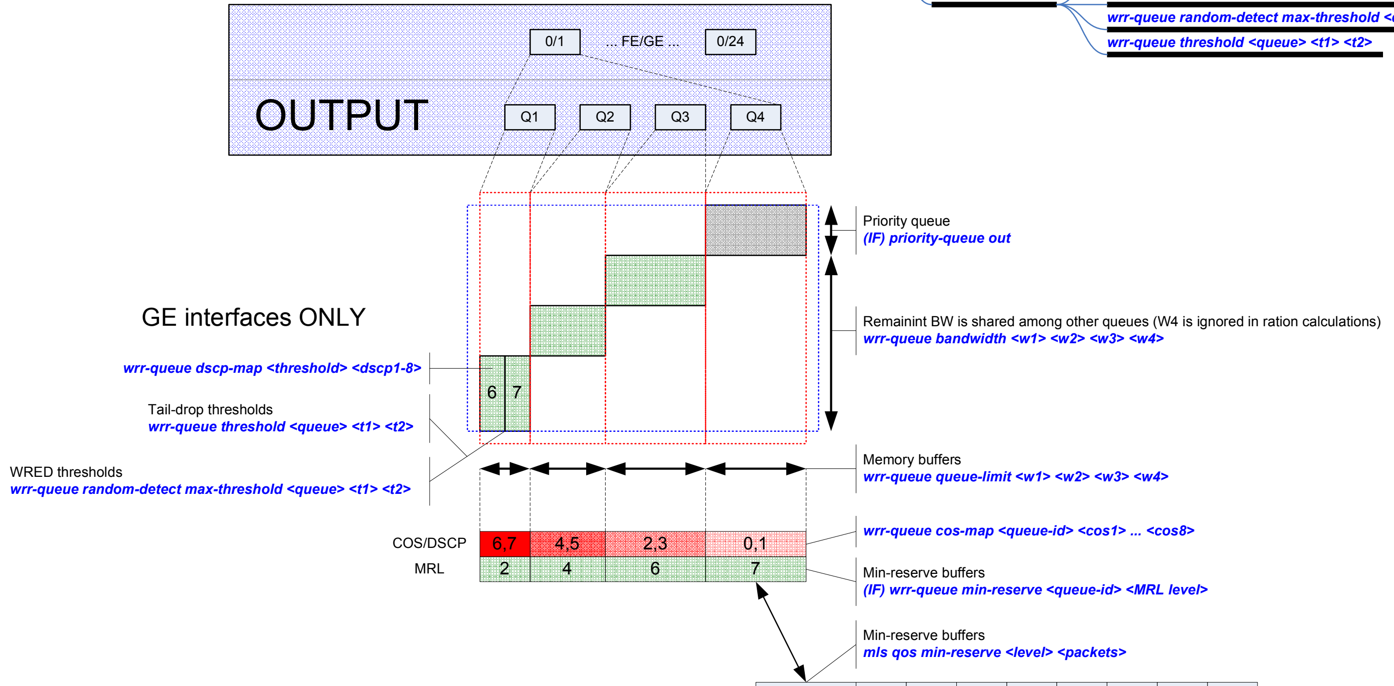
**2.** Mapping CoS Values to Select Egress Queues
- ***wrr-queue cos-map <queue-id> <cos1> ... <cos8>***

**3.** Allocating Bandwidth among Egress Queues
- ***wrr-queue bandwidth <w1> <w2> <w3> <w4>***
  Ratio of weights is the ratio of frequency in which WRR scheduler dequeues packets from each queue

Egress Queue Size Ratios
- ***wrr-queue queue-limit <w1> <w2> <w3> <w4>***
  Relative size difference in the numbers show the relative differences in the queue sizes

Enable expedite queue
- ***priority-queue out***
  WRR weight and queue size ratios are affected because there is one fewer queue participating in WRR. This means that weight4 in the ***wrr-queue bandwidth*** command is ignored (not used in the ratio calculation)

WRED on GE ports
- Each Q has 2 thresholds defined as % of Q len. Linear drop between T1 and T2 from 0 to 100%
- ***wrr-queue dscp-map <threshold> <dscp> ...***
  By default all 64 DSCPs are mapped to T1
- ***wrr-queue random-detect max-threshold <queue> <t1> <t2>***
- ***wrr-queue threshold <queue> <t1> <t2>***



GE interfaces ONLY

***wrr-queue dscp-map <threshold> <dscp1-8>***

Tail-drop thresholds
***wrr-queue threshold <queue> <t1> <t2>***

WRED thresholds
***wrr-queue random-detect max-threshold <queue> <t1> <t2>***

Priority queue
***(IF) priority-queue out***

Remainint BW is shared among other queues (W4 is ignored in ration calculations)
***wrr-queue bandwidth <w1> <w2> <w3> <w4>***

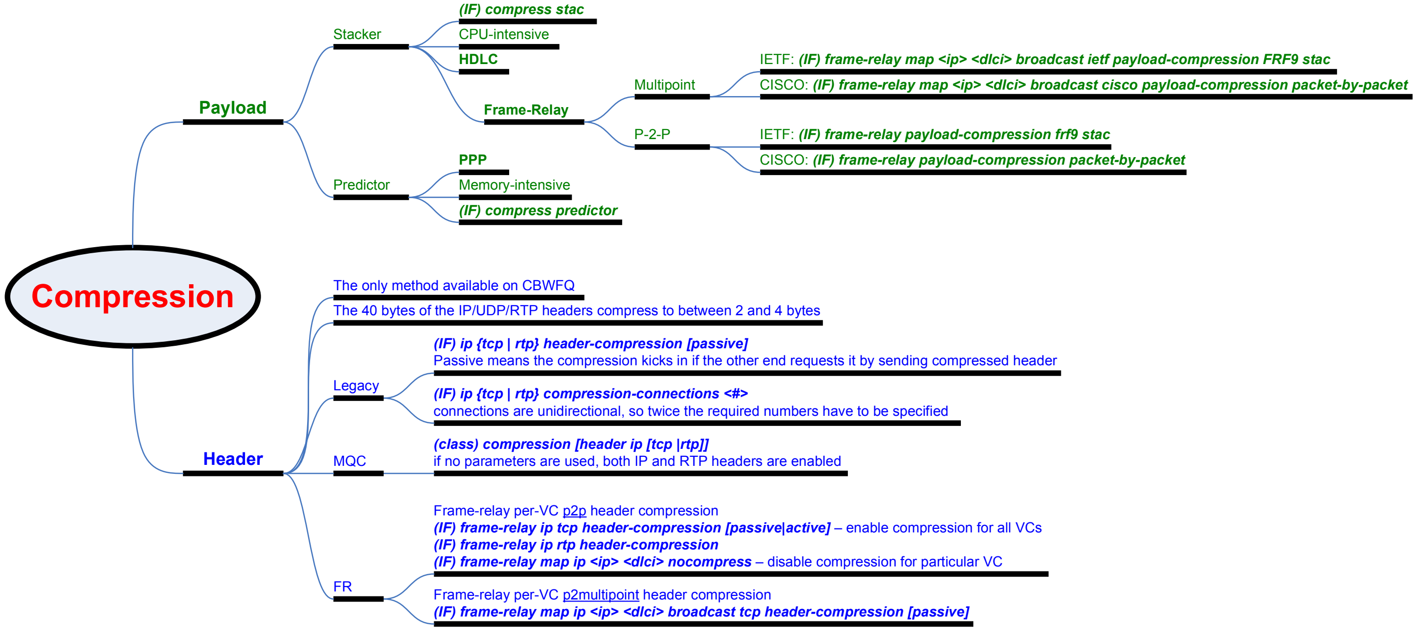
Memory buffers
***wrr-queue queue-limit <w1> <w2> <w3> <w4>***

***wrr-queue cos-map <queue-id> <cos1> ... <cos8>***

Min-reserve buffers
***(IF) wrr-queue min-reserve <queue-id> <MRL level>***

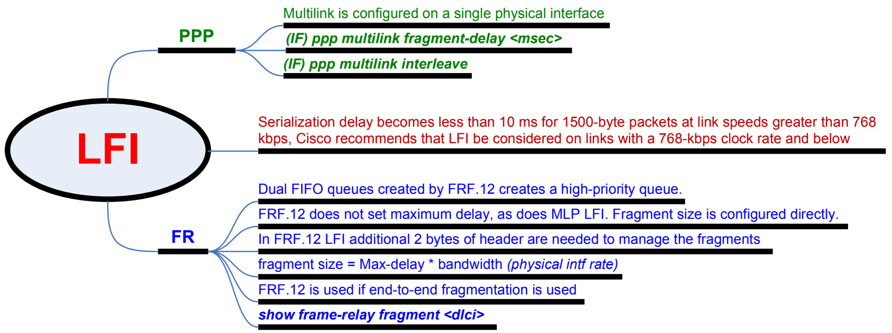
Min-reserve buffers
***mls qos min-reserve <level> <packets>***

| MRL | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Buffer size | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 |



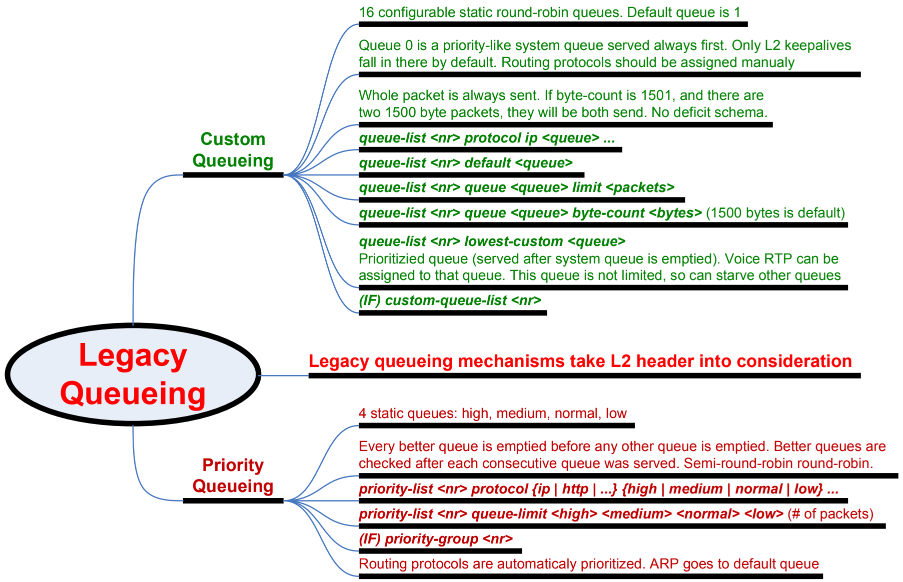
**Compression**

- **Payload**
  - Stacker
    - *(IF) compress stac*
    - CPU-intensive
    - **HDLC**
    - **Frame-Relay**
      - Multipoint
        - IETF: ***(IF) frame-relay map <ip> <dlci> broadcast ietf payload-compression FRF9 stac***
        - CISCO: ***(IF) frame-relay map <ip> <dlci> broadcast cisco payload-compression packet-by-packet***
      - P-2-P
        - IETF: ***(IF) frame-relay payload-compression frf9 stac***
        - CISCO: ***(IF) frame-relay payload-compression packet-by-packet***
  - Predictor
    - **PPP**
    - Memory-intensive
    - ***(IF) compress predictor***
- **Header**
  - The only method available on CBWFQ
  - The 40 bytes of the IP/UDP/RTP headers compress to between 2 and 4 bytes
  - Legacy
    - ***(IF) ip {tcp | rtp} header-compression [passive]***<br>Passive means the compression kicks in if the other end requests it by sending compressed header
    - ***(IF) ip {tcp | rtp} compression-connections <#>***<br>connections are unidirectional, so twice the required numbers have to be specified
  - MQC
    - ***(class) compression [header ip [tcp |rtp]]***<br>if no parameters are used, both IP and RTP headers are enabled
  - FR
    - Frame-relay per-VC p2p header compression<br>***(IF) frame-relay ip tcp header-compression [passive|active]*** – enable compression for all VCs<br>***(IF) frame-relay ip rtp header-compression***<br>***(IF) frame-relay map ip <ip> <dlci> nocompress*** – disable compression for particular VC
    - Frame-relay per-VC p2multipoint header compression<br>***(IF) frame-relay map ip <ip> <dlci> broadcast tcp header-compression [passive]***



**LFI**

- **PPP**
  - 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
- **FR**
  - Dual FIFO queues created by FRF.12 creates a high-priority queue.
  - FRF.12 does not set maximum delay, as does MLP LFI. Fragment size is configured directly.
  - In FRF.12 LFI additional 2 bytes of header are needed to manage the fragments
  - fragment size = Max-delay * bandwidth *(physical intf rate)*
  - FRF.12 is used if end-to-end fragmentation is used
  - ***show frame-relay fragment <dlci>***



**Legacy Queueing**

- **Custom Queueing**
  - 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 manualy
  - Whole packet is always sent. If byte-count is 1501, and there are two 1500 byte packets, they will be both send. No deficit schema.
  - ***queue-list <nr> protocol ip <queue> ...***
  - ***queue-list <nr> default <queue>***
  - ***queue-list <nr> queue <queue> limit <packets>***
  - ***queue-list <nr> queue <queue> byte-count <bytes>*** (1500 bytes is default)
  - ***queue-list <nr> lowest-custom <queue>***<br>Prioritizied 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>***
- **Legacy queueing mechanisms take L2 header into consideration**
- **Priority Queueing**
  - 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. Semi-round-robin round-robin.
  - ***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 automaticaly prioritized. ARP goes to default queue

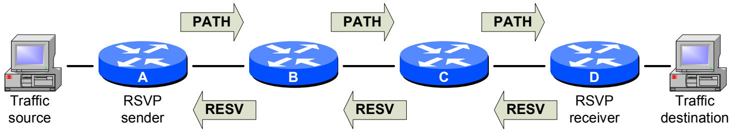
# RSVP

## Features

- Core of integrated services (end-to-end QOS model)
- Poorly scalable – each flow requires own reservation. Used mainly for MPLS Traffic Engineering
- Flows are unidirectional, so each side has to request own RSVP path
- Traffic exceeding reservation is treated as a best-effort
- RSVP reservations take precedence over user-defined classed in CBWFQ

## 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 requests 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. RESV 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



## RESV

- Flowspec
  - Rspec
    - Reservation specification (class of service requested)
  - Tspec
    - Traffic specification (parameters for traffic metering – Avg rate and burst)
- Filterspec
  - Contains sources which may use reservation installed by the receiver
  - FF – fixed filter – only one cource can use the reservation with specific Tspec parameters
  - SE – Shared explicit filter – multiple, but explicitly defined sources can use the reservation (receiver specifies sources' IPs)
  - WF – Wildcasr filter – any sender can use the reservation

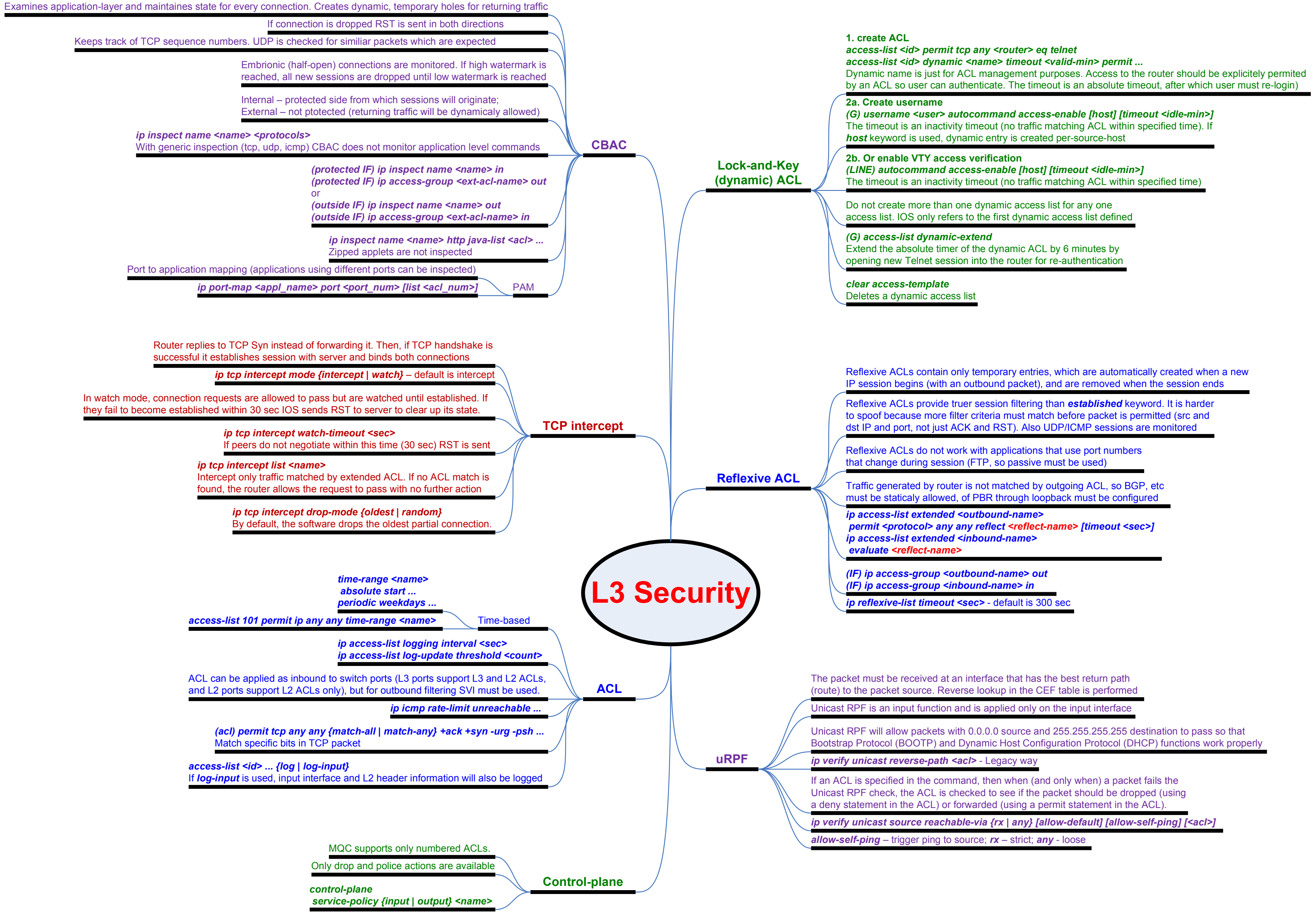
## Configuration

- ***ip rsvp bandwidth [<total bw kbps> <single flow kbps>]***
  By default 75% ow BW can be reserved
- If RSVP BW is configured on subintf it must be also configured on main interface as a sum of all subintf BW values
- Fair-queueing is required. FRTS disables fair-queueing on intf, so it must be added to the FR class (***frame-relay fair-queue***)
- RSVP BW is substracted from interface bandwidth available for CBWFQ.
- Proxy – if connected client is not RSVP-aware
  - ***ip rsvp sender ...***
  - ***ip rsvp reservation ...***
- LLQ
  - PQ profile defines parameters 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 itself (priority queue) is not required in CBWFQ
  - ***ip rsvp pq-profile <max-rate> <max-burst> <peak-to-avg ratio in %>***

## Testing

- ***ip rsvp sender-host <rcv IP> <snd IP> {tcp | udp | ip} <dst port> <src port> <session bw kbps> <burst kbps>***
  RSVP PATH signalling can be tested with this command
- ***show ip rsvp sender***
- ***ip rsvp reservation-host <rcv IP> <snd IP> {tcp | udp | ip} <dst port> <src port> {FF | SE | WF} <session bw kbps> <burst kbps>***
  RSVP RESV signalling can be tested with (FF – fixed filter for single reservation, SE – shared explicit with limited scope, WF – wildcard filter with unlimited scope)
- ***show ip rsvp reservation***
- ***show ip rsvp installed [detail]***

**Packets initiated by a router are not matched by outbound ACL or any inspection !!!**



# L3 Security

## CBAC

Examines application-layer and maintaines state for every connection. Creates dynamic, temporary holes for returning traffic

If connection is dropped RST is sent in both directions

Keeps track of TCP sequence numbers. UDP is checked for similiar packets which are expected

Embrionic (half-open) connections are monitored. If high watermark is reached, all new sessions are dropped until low watermark is reached

Internal – protected side from which sessions will originate;
External – not ptotected (returning traffic will be dynamicaly allowed)

***ip inspect name <name> <protocols>***
With generic inspection (tcp, udp, icmp) CBAC does not monitor application level commands

***(protected IF) ip inspect name <name> in***
***(protected IF) ip access-group <ext-acl-name> out***
or
***(outside IF) ip inspect name <name> out***
***(outside IF) ip access-group <ext-acl-name> in***

***ip inspect name <name> http java-list <acl> ...***
Zipped applets are not inspected

PAM
- Port to application mapping (applications using different ports can be inspected)
- ***ip port-map <appl_name> port <port_num> [list <acl_num>]***

## 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

***ip tcp intercept mode {intercept | watch}*** – default is intercept

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 its state.

***ip tcp intercept watch-timeout <sec>***
If peers do not negotiate within this time (30 sec) RST is sent

***ip tcp intercept list <name>***
Intercept only traffic matched by extended ACL. If no ACL match is found, the router allows the request to pass with no further action

***ip tcp intercept drop-mode {oldest | random}***
By default, the software drops the oldest partial connection.

## ACL

Time-based
- ***time-range <name>***
  ***absolute start ...***
  ***periodic weekdays ...***
- ***access-list 101 permit ip any any time-range <name>***

***ip access-list logging interval <sec>***
***ip access-list log-update threshold <count>***

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 filtering SVI must be used.

***ip icmp rate-limit unreachable ...***

***(acl) permit tcp any any {match-all | match-any} +ack +syn -urg -psh ...***
Match specific bits in TCP packet

***access-list <id> ... {log | log-input}***
If ***log-input*** is used, input interface and L2 header information will also be logged

## Control-plane

MQC supports only numbered ACLs.

Only drop and police actions are available

***control-plane***
***service-policy {input | output} <name>***

## Lock-and-Key (dynamic) ACL

**1. create ACL**
***access-list <id> permit tcp any <router> eq telnet***
***access-list <id> dynamic <name> timeout <valid-min> permit ...***
Dynamic name is just for ACL management purposes. Access to the router should be explicitely permited by an ACL so user can authenticate. The timeout is an absolute timeout, after which user must re-login)

**2a. Create username**
***(G) username <user> autocommand access-enable [host] [timeout <idle-min>]***
The timeout is an inactivity timeout (no traffic matching ACL within specified time). If ***host*** keyword is used, dynamic entry is created per-source-host

**2b. Or enable VTY access verification**
***(LINE) autocommand access-enable [host] [timeout <idle-min>]***
The timeout is an inactivity timeout (no traffic matching ACL within specified time)

Do not create more than one dynamic access list for any one access list. IOS only refers to the first dynamic access list defined

***(G) access-list dynamic-extend***
Extend the absolute timer of the dynamic ACL by 6 minutes by opening new Telnet session into the router for re-authentication

***clear access-template***
Deletes a dynamic access list

## Reflexive ACL

Reflexive ACLs contain only temporary entries, which are automatically created when a new IP session begins (with an outbound packet), and are removed when the session ends

Reflexive ACLs provide truer session filtering than ***established*** keyword. It is harder to spoof because more filter criteria must match before packet is permitted (src and dst IP and port, not just ACK and RST). Also UDP/ICMP sessions are monitored

Reflexive ACLs do not work with applications that use port numbers that change during session (FTP, so passive must be used)

Traffic generated by router is not matched by outgoing ACL, so BGP, etc must be staticaly allowed, of PBR through loopback must be configured

***ip access-list extended <outbound-name>***
 ***permit <protocol> any any reflect <reflect-name> [timeout <sec>]***
***ip access-list extended <inbound-name>***
 ***evaluate <reflect-name>***

***(IF) ip access-group <outbound-name> out***
***(IF) ip access-group <inbound-name> in***

***ip reflexive-list timeout <sec>*** - default is 300 sec

## uRPF

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

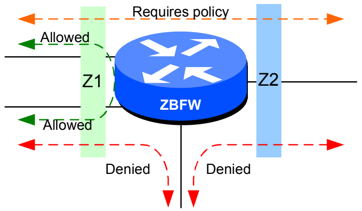
***ip verify unicast reverse-path <acl>*** - Legacy way

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).

***ip verify unicast source reachable-via {rx | any} [allow-default] [allow-self-ping] [<acl>]***

***allow-self-ping*** – trigger ping to source; ***rx*** – strict; ***any*** - loose

# Zone-based Policy FW



## Zones

- 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
- Self-zone is router itself. Traffic cannot be policed
- When ZBFW is configured all the interfaces must be a member of one security zone or another. No traffic will pass to an interface which is not assigned to any 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
- A zone-pair allows to specify a **unidirectional** firewall policy between two security zones. However it is not required to define policy for returning traffic, which is allowed by a statefull firewall operation
- 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
- ACLs applied to interfaces that are members of zones are processed before the policy is applied on the zone-pair
- ***(G) zone security <name>*** ! create a zone
- ***zone-pair security <pair-name> {source <zone-name> | self] destination [self | <zone-name>]***  
  ***service-policy type inspect <map-name>*** ! if policy map is not applied, traffic is dropped by default
- ***(IF) zone-member security <zone-name>***

## Inspection

- Inspection can be configured per-flow. Not all traffic flowing through an interface must be inspected
- Inspection configuration is based on class-map (***type inspect***), policy-map, service-policy, just like in QoS
- Application inspection
  - 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, Gnutella, H.323, HTTP, Kazaa, ICQ, MSN IM, POP3, SIP, SMTP, SunRPC
  - NBAR is not available for bridged packets (transparent firewall between bridged interface)
  - ***(G) class-map type inspect <protocol> [match-any | match-all] <name>***  
    ***(G) policy-map type inspect <protocol> <name>***
  - ***(G) parameter-map type urlfpolicy {local | n2h2 | websense} <name>***  
    ***(G) class-map type urlfilter {<name> | {n2h2 | websense} <name>}***  
    ***(G) policy-map type inspect urlfilter <name>***

## Class map

- ***(G) class-map type inspect [match-any | match-all] <name>***  
  Creates a Layer 3 or Layer 4 inspect type class map
- ***match class-map <name>***  
  Classes can be used to define hierarchical match
- ***match protocol <name> [signature]***  
  Only Cisco IOS stateful packet inspection supported protocols can be used as match criteria in inspect type class maps. Signature-based p2p packets can be matched
- ***match access-group {<acl> | name <acl-name>}***  
  Match based on the ACL name or number

## Policy map

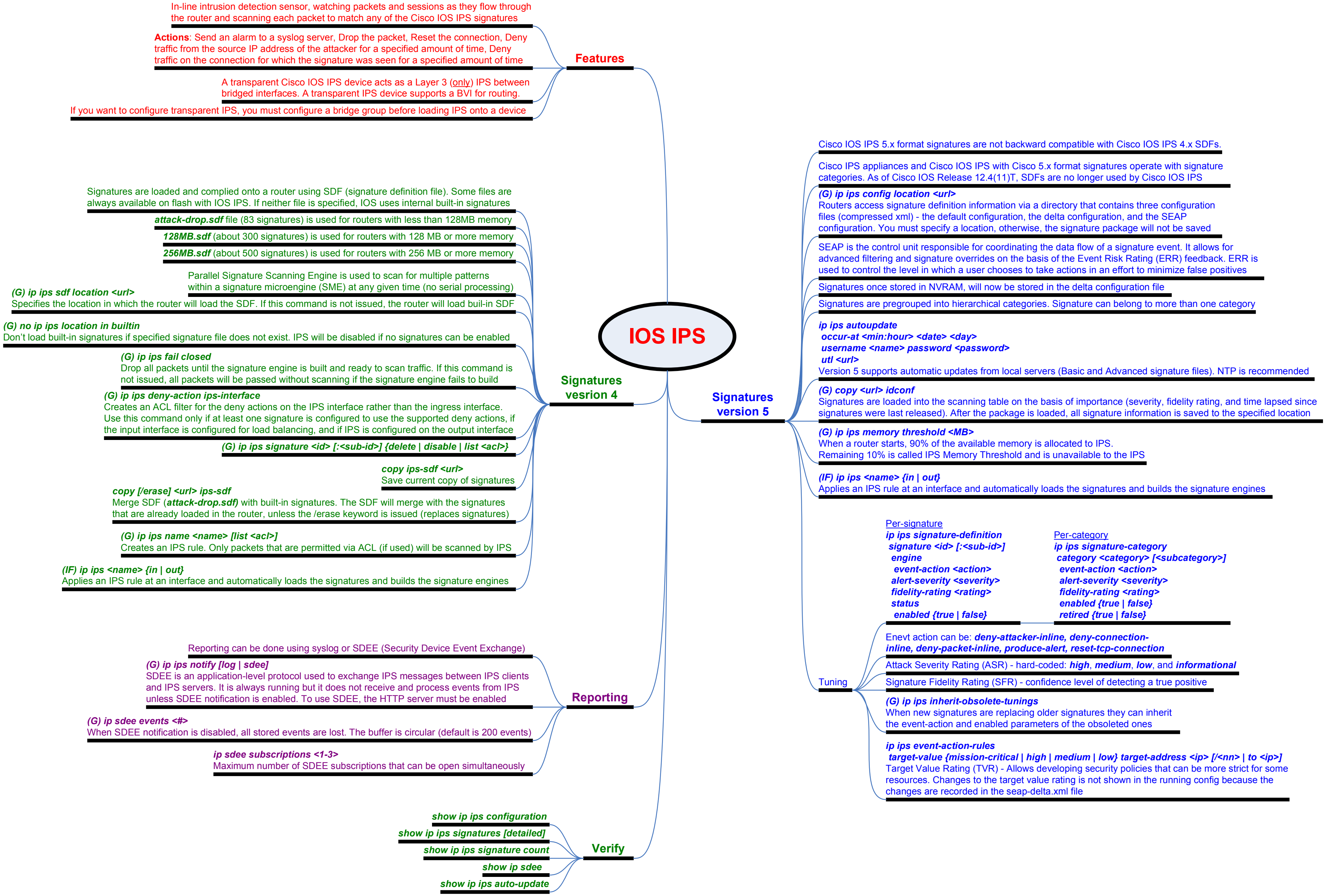
- ***policy-map type inspect <name>***  
  ***class type inspect <name>***  
  Creates a Layer 3 and Layer 4 inspect type policy map
- The policy map can include class maps only of the same type
- There is always a class-default at the end. Default action is ***drop.*** It can be changed to ***inspect***
- ***police rate <bps> burst <size>***  
  Policing (rate-limiting) can only be specified in L3/L4 policy maps. Inspection must be enabled.
- ***drop [log]***  
  Drop packets
  - ***pass***  
    Allow packets
- ***service-policy type inspect <name>***  
  There can be a maximum of two levels in a hierarchical inspect service-policy.  Parameters in the lower levels override those in the top levels
- ***urlfilter <parameter-map-name>***  
  Enables Cisco IOS firewall URL filtering
- ***inspect [<parameter-map-name>]***  
  Enables Cisco IOS stateful packet inspection

## Verify

- ***show policy-map type inspect zone-pair session***

## Parameter maps

- Inspect
  - ***parameter-map type inspect <name>***
  - ***alert {on | off}*** ! Alert messages are displayed on the console  
    ***audit-trail {on | off}***
  - ***dns-timeout <sec>***
  - ***max-incomplete {low <#> | high <#>}***
  - ***sessions maximum <#>***
  - ***one-minute {low <#> | high <#>}***
  - ***tcp finwait-time <sec>***
  - ***{tcp | udp | icmp} idle-time <sec>***
  - ***tcp max-incomplete host <threshold> [block-time <minutes>]***
  - ***tcp synwait-time <sec>***
  - ***tcp window-scale-enforcement loose***  
    Disables the window scale option check
- URL filter
  - ***parameter-map type urlfilter <name>*** - hidden since 12.4(20)T
  - ***allow-mode {on | off}***  
    Turns on or off the default mode of the filtering algorithm
  - ***cache <#>***  
    Controls how the URL filter handles the cache it maintains of HTTP servers
  - ***exclusive-domain {deny | permit} <domain-name>***  
    Firewall does not send DNS request for traffic destined for those domains
  - ***max-request <#>***
  - ***max-resp-pak <number-of-requests>***  
    Maximum number of HTTP responses that the Cisco IOS firewall can keep in its packet buffer
  - ***server vendor {n2h2 | websense} {<ip> | <hostname> [port <#>]} [outside] [log] [retrans <count>] [timeout <sec>]***  
    Specifies the URL filtering server
  - ***source-interface <if>***
- Protocol specific
  - ***parameter-map type protocol-info <name>***  
    Protocol-specific parameter maps can be created only for Instant Messenger applications
  - ***server {name <string> [snoop] | ip {<ip> | range <start> <end>}***  
    This command can be defined multiple times to match many servers
- Out-of-Order
  - Default behaviour is to drop packets arriving out of order
  - OoO packet processing is enabled by default when a L7 policy is configured for DPI
  - Not supported in SMTP, as SMTP supports masking action that requires packet modification
  - ***parameter-map type ooo global***  
    OoO paramter map defines global operations for all interfaces
  - ***tcp reassembly alarm {on | off}***
  - ***tcp reassembly memory limit <limit>***  
    OoO buffer size
  - ***tcp reassembly queue length <#>***  
    OoO queue
  - ***tcp reassembly timeout <sec>***
- ***clear zone-pair inspect sessions***  
  Changes to the parameter map are not reflected on connections already established through the firewall



# IOS IPS

## Features

- 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 a BVI for routing.
- If you want to configure transparent IPS, you must configure a bridge group before loading IPS onto a device

## Signatures vesrion 4

- Signatures are loaded and complied onto a router using SDF (signature definition file). Some files are always available on flash with IOS IPS. If neither file is specified, IOS uses internal built-in signatures
- ***attack-drop.sdf*** file (83 signatures) is used for routers with less than 128MB memory
- ***128MB.sdf*** (about 300 signatures) is used for routers with 128 MB or more memory
- ***256MB.sdf*** (about 500 signatures) is used for routers with 256 MB or more memory
- Parallel Signature Scanning Engine is used to scan for multiple patterns within a signature microengine (SME) at any given time (no serial processing)
- ***(G) ip ips sdf location <url>***
  Specifies the location in which the router will load the SDF. If this command is not issued, the router will load buil-in SDF
- ***(G) no ip ips location in builtin***
  Don't load built-in signatures if specified signature file does not exist. IPS will be disabled if no signatures can be enabled
- ***(G) ip ips fail closed***
  Drop all packets until the signature engine is built and ready to scan traffic. If this command is not issued, all packets will be passed without scanning if the signature engine fails to build
- ***(G) ip ips deny-action ips-interface***
  Creates an ACL filter for the deny actions on the IPS interface rather than the ingress interface. Use this command only if at least one signature is configured to use the supported deny actions, if the input interface is configured for load balancing, and if IPS is configured on the output interface
- ***(G) ip ips signature <id> [:<sub-id>] {delete | disable | list <acl>}***
- ***copy ips-sdf <url>***
  Save current copy of signatures
- ***copy [/erase] <url> ips-sdf***
  Merge SDF (***attack-drop.sdf)*** with built-in signatures. The SDF will merge with the signatures that are already loaded in the router, unless the /erase keyword is issued (replaces signatures)
- ***(G) ip ips name <name> [list <acl>]***
  Creates an IPS rule. Only packets that are permitted via ACL (if used) will be scanned by IPS
- ***(IF) ip ips <name> {in | out}***
  Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines

## Reporting

- Reporting can be done using syslog or SDEE (Security Device Event Exchange)
- ***(G) ip ips notify [log | sdee]***
  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
- ***(G) ip sdee events <#>***
  When SDEE notification is disabled, all stored events are lost. The buffer is circular (default is 200 events)
- ***ip sdee subscriptions <1-3>***
  Maximum number of SDEE subscriptions that can be open simultaneously

## Verify

- ***show ip ips configuration***
- ***show ip ips signatures [detailed]***
- ***show ip ips signature count***
- ***show ip sdee***
- ***show ip ips auto-update***

## Signatures version 5

- Cisco IOS IPS 5.x format signatures are not backward compatible with Cisco IOS IPS 4.x SDFs.
- Cisco IPS appliances and Cisco IOS IPS with Cisco 5.x format signatures operate with signature categories. As of Cisco IOS Release 12.4(11)T, SDFs are no longer used by Cisco IOS IPS
- ***(G) ip ips config location <url>***
  Routers access signature definition information via a directory that contains three configuration files (compressed xml) - the default configuration, the delta configuration, and the SEAP configuration. You must specify a location, otherwise, the signature package will not be saved
- SEAP is the control unit responsible for coordinating the data flow of a signature event. It allows for advanced filtering and signature overrides on the basis of the Event Risk Rating (ERR) feedback. ERR is used to control the level in which a user chooses to take actions in an effort to minimize false positives
- Signatures once stored in NVRAM, will now be stored in the delta configuration file
- Signatures are pregrouped into hierarchical categories. Signature can belong to more than one category
- ***ip ips autoupdate***
  ***occur-at <min:hour> <date> <day>***
  ***username <name> password <password>***
  ***utl <url>***
  Version 5 supports automatic updates from local servers (Basic and Advanced signature files). NTP is recommended
- ***(G) copy <url> idconf***
  Signatures are loaded into the scanning table on the basis of importance (severity, fidelity rating, and time lapsed since signatures were last released). After the package is loaded, all signature information is saved to the specified location
- ***(G) ip ips memory threshold <MB>***
  When a router starts, 90% of the available memory is allocated to IPS.
  Remaining 10% is called IPS Memory Threshold and is unavailable to the IPS
- ***(IF) ip ips <name> {in | out}***
  Applies an IPS rule at an interface and automatically loads the signatures and builds the signature engines
- Tuning
  - Per-signature
    ***ip ips signature-definition***
    ***signature <id> [:<sub-id>]***
    ***engine***
    ***event-action <action>***
    ***alert-severity <severity>***
    ***fidelity-rating <rating>***
    ***status***
    ***enabled {true | false}***
  - Per-category
    ***ip ips signature-category***
    ***category <category> [<subcategory>]***
    ***event-action <action>***
    ***alert-severity <severity>***
    ***fidelity-rating <rating>***
    ***enabled {true | false}***
    ***retired {true | false}***
  - Enevt action can be: ***deny-attacker-inline, deny-connection-inline, deny-packet-inline, produce-alert, reset-tcp-connection***
  - Attack Severity Rating (ASR) - hard-coded: ***high***, ***medium***, ***low***, and ***informational***
  - Signature Fidelity Rating (SFR) - confidence level of detecting a true positive
  - ***(G) ip ips inherit-obsolete-tunings***
    When new signatures are replacing older signatures they can inherit the event-action and enabled parameters of the obsoleted ones
  - ***ip ips event-action-rules***
    ***target-value {mission-critical | high | medium | low} target-address <ip> [/<nn> | to <ip>]***
    Target Value Rating (TVR) - Allows developing security policies that can be more strict for some resources. Changes to the target value rating is not shown in the running config because the changes are recorded in the seap-delta.xml file

# L2 security

## DHCP snooping

- ***ip dhcp snooping***
- ***ip dhcp snooping vlan <#>***
- ***(IF) ip dhcp snooping trust***
  Enable ports with trusted devices (DHCP server)
- 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>***
  By default all entries are removed if switch is reloaded. Dynamic and static entries can be stored in external database.
- ***(IF) ip dhcp snooping limit rate <#>***
- ***no ip dhcp relay information option***
  Disable (enabled by default) the switch to insert and remove DHCP relay information (option-82 field). Option-82 adds circuit-id (port to which host is connected) and remote-id (BID of switch where host is connected). Switch adds those options to DHCP Discovery message sent by host. Must be enabled on each switch. It is informational field used by DHCP server to assign IPs. If option-82 is added, giaddr is set to 0, what is rejected by Cisco IOS DHCP server.
- ***(G) ip dhcp relay information trust-all***
  ***(IF) ip dhcp relay information trusted***
  set on DHCP server to trust all messages (accept messages with option-82 – giaddr=0)
- ***(G) ip dhcp snooping information option allow-untrusted***
  Not recommended if any untrusted devices are connected to the switch
- ***(IF) ip dhcp snooping vlan <id> information option …***
  ***(G) ip dhcp snooping information option …***
  Configured option-82 fields (ciscuit-id, type) per-interface or globaly
- ***ip dhcp snooping binding <MAC> vlan <id> <ip> interface <if> expiry <sec>***
  Configured in privilege mode, not config mode. Not saved to NVRAM.

## IP source guard

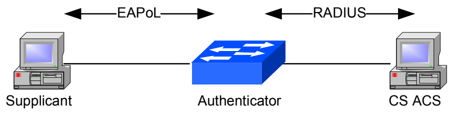
- DHCP snooping extension used to prevent attacks when a host tries to use neighbor's IP
- Checks source IP of received packet against DHCP binding table
- DHCP snooping must be enabled on the access VLAN to which the interface belongs
- ***(IF) ip verify source [port-security]***
  By default L3 is checked (user can change MAC), but if used with port-security L2 and L3 is checked
- ***ip source binding <MAC> vlan <id> <ip> interface <if>***
  This is configured in global mode, so it's stored in NVRAM, unlike DHCP snooping DB

## Port blocking

- Prevent unknown unicast or multicast traffic from being forwarded from one port to another
- ***(IF) switchport block {unicast | multicast}***

## 802.1x

- EAP provides link layer security framework. It can run on any data link(802, PPP)
- ***(G) dot1x system auth-control***
  Enable dot1x (required)
- ***aaa authentication dot1x group …***
  Enable ***aaa new-model*** and define authentication method for dot1x requests
- ***(IF) dot1x port-control {auto | force-authorized | force-unauthorized}***
  Only auto mode generated 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.
- ***dot1x guest-vlan <vlan-id>***
  The switch assigns clients to a guest VLAN when it does not receive a response to EAPOL
- ***(IF) dot1x host-mode multi-host***
  Allows all hosts connected to one port to use authentication performed only by one host
- ***dot1x auth-fail vlan <vlan-id>***
  Define restricted vlan upon authentication failure. The user is not notified of the authentication failure.
- ***dot1x reauthentication [interface <intf>]***
  Re-enable authentication on restricted vlan (exec mode)
- ***dot1x timeout reauth-period <sec>***
  Re-authentication period for restricted vlan



## Dynamic ARP inspection

- In non-DHCP environments, dynamic ARP inspection can validate ARP packets against user-configured ARP access control lists (ACLs) for hosts with statically configured IP addresses
- ***(G) ip arp inspection vlan <#>***
- ***(IF) ip arp inspection trust -*** Define trusted interface
- ***arp access-list <acl-name>***
  ***permit ip host <sender-ip> mac host <sender-mac> [log]***
  At least two entries are required, one for each host.
- ***ip arp inspection filter <ARP-acl> vlan <range> [static]***
  DHCP snooping is not required if ***static*** keyword is used. Otherwise, ACL is checked first, then DHCP
- ***ip arp inspection validate [src-mac] [dst-mac] [ip]***
- ***ip arp inspection limit {rate <pps> [burst <intv>] | none}*** (default 15pps/1sec)
- ***ip arp inspection log-buffer {entries <#> | logs <#> interval <sec>}***

## Port security

- ***switchport port-security*** – enable port security feature
- ***switchport port-security maximum <#> [vlan {voice | access}]***
  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 {<id> | access | voice}*** – static MAC address
- ***switchport port-security mac-address sticky***
  remember first MAC learned. MAC is added to configuration, but config is not automaticaly saved to nvram. If you configure fewer static MACs than the allowed max, the remaining dynamically learned MACs will be converted to sticky
- ***switchport port-security violation {protect | restrict | shutdown}***
  Protect - packets with unknown source addresses are dropped. Restrict – like protect, but you are notified that a security violation has occurred. Shutdown – interface is error-disabled (default)
- ***switchport port-security aging {static | time time | type {absolute | inactivity}}***
- ***snmp-server enable traps port-security trap-rate <#/sec>***

## Storm control

- When rate of mcast traffic exceeds a threshold, all incoming traffic (broadcast, multicast, and unicast) is dropped. Only spanning-tree packets are forwarded. When bcast and ucast thresholds are exceeded, traffic is blocked for only the type of traffic that exceeded the threshold.
- ***storm-control { broadcast | multicast | unicast } level pps <high> [<low>]***
- ***storm-control action {shutdown | trap}***

## Protected port

- Ensures that there is no exchange of ucast, bcast, or mcast traffic between ports on the switch
- All data traffic passing between protected ports must be forwarded through a Layer 3 device. ICMP redirects are automaticaly disabled on protected ports.
- Forwarding between a protected port and a non-protected port proceeds as usual
- Does not span across switches. Blocks L2, but ping 255.255.255.255 will reach hosts (port blockinng must be used to block unnown unicasts and broadcasts)
- ***(IF) switchport protected***

## VLAN ACL

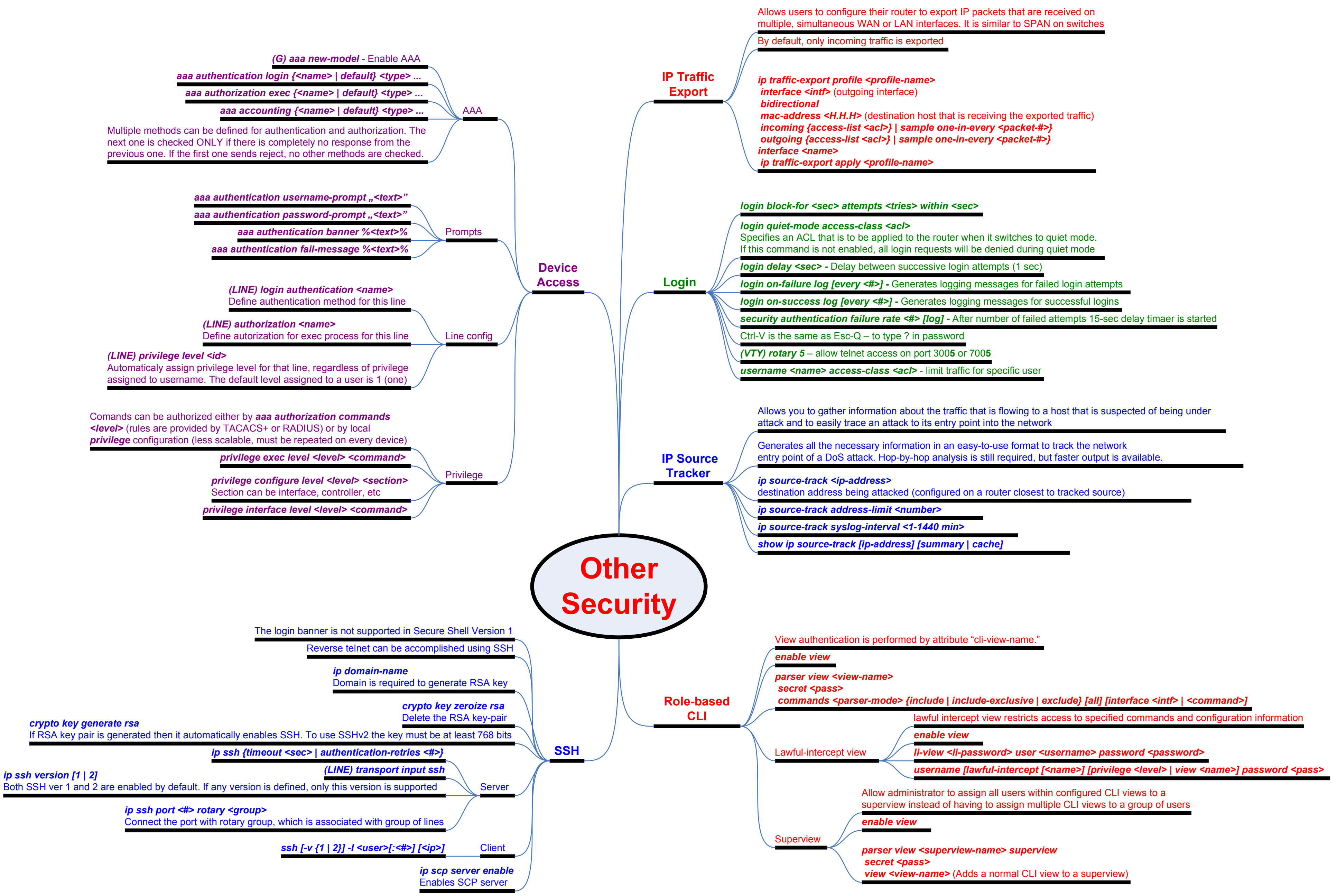
- VLAN ACLs are inbound and they can conflict with other per-port filters
- VLAN ACLs run in hardware. They must be re-applied if changed
- ***vlan access-map <name> <seq>*** (access-map is like route-map, many entries with different actions)
  ***match {ip | mac} address <acl>***
  ***action {drop | forward}***
  ***vlan filter <name> vlan-list <vlans>***

## Static MAC

- ***mac-address-table static 0000.1111.1111 vlan <vlan> interface <if>***
- ***mac-address-table static 0000.1111.1111 vlan <vlan> drop*** - src and dst MAC will be dropped

## 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 aarp***
  ***permit any any***
  ***interface fastethernet 0/0***
  ***mac access-group <name> in*** (Always IN)



# Other Security

## Device Access

### AAA

***(G) aaa new-model*** - Enable AAA

***aaa authentication login {<name> | default} <type> ...***

***aaa authorization exec {<name> | default} <type> ...***

***aaa accounting {<name> | default} <type> ...***

Multiple methods can be defined for authentication and authorization. The next one is checked ONLY if there is completely no response from the previous one. If the first one sends reject, no other methods are checked.

### Prompts

***aaa authentication username-prompt „<text>”***

***aaa authentication password-prompt „<text>”***

***aaa authentication banner %<text>%***

***aaa authentication fail-message %<text>%***

### Line config

***(LINE) login authentication <name>***
Define authentication method for this line

***(LINE) authorization <name>***
Define autorization for exec process for this line

***(LINE) privilege level <id>***
Automaticaly assign privilege level for that line, regardless of privilege assigned to username. The default level assigned to a user is 1 (one)

### Privilege

Comands can be authorized either by ***aaa authorization commands <level>*** (rules are provided by TACACS+ or RADIUS) or by local ***privilege*** configuration (less scalable, must be repeated on every device)

***privilege exec level <level> <command>***

***privilege configure level <level> <section>***
Section can be interface, controller, etc

***privilege interface level <level> <command>***

## SSH

The login banner is not supported in Secure Shell Version 1

Reverse telnet can be accomplished using SSH

***ip domain-name***
Domain is required to generate RSA key

***crypto key zeroize rsa***
Delete the RSA key-pair

***crypto key generate rsa***
If RSA key pair is generated then it automatically enables SSH. To use SSHv2 the key must be at least 768 bits

### Server

***ip ssh {timeout <sec> | authentication-retries <#>}***

***(LINE) transport input ssh***

***ip ssh version [1 | 2]***
Both SSH ver 1 and 2 are enabled by default. If any version is defined, only this version is supported

***ip ssh port <#> rotary <group>***
Connect the port with rotary group, which is associated with group of lines

### Client

***ssh [-v {1 | 2}] -l <user>[:<#>] [<ip>]***

***ip scp server enable***
Enables SCP server

## 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-#>}
 outgoing {access-list <acl>} | sample one-in-every <packet-#>}
interface <name>
 ip traffic-export apply <profile-name>
```

## Login

***login block-for <sec> attempts <tries> within <sec>***

***login quiet-mode access-class <acl>***
Specifies an ACL that is to be applied to the router when it switches to quiet mode.
If this command is not enabled, all login requests will be denied during quiet mode

***login delay <sec> -*** Delay between successive login attempts (1 sec)

***login on-failure log [every <#>] -*** Generates logging messages for failed login attempts

***login on-success log [every <#>] -*** Generates logging messages for successful logins

***security authentication failure rate <#> [log] -*** After number of failed attempts 15-sec delay timaer is started

Ctrl-V is the same as Esc-Q – to type ? in password

***(VTY) rotary 5*** – allow telnet access on port 300**5** or 700**5**

***username <name> access-class <acl>*** - limit traffic for specific user

## IP Source Tracker

Allows you 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

Generates all the necessary information in an easy-to-use format to track the network entry point of a DoS attack. Hop-by-hop analysis is still required, but faster output is available.

***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} [all] [interface <intf> | <command>]
```

### Lawful-intercept view

lawful intercept view restricts access to specified commands and configuration information

***enable view***

***li-view <li-password> user <username> password <password>***

***username [lawful-intercept [<name>] [privilege <level> | view <name>] password <pass>***

### Superview

Allow 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> superview
 secret <pass>
 view <view-name> (Adds a normal CLI view to a superview)
```

# MPLS Control & Forwarding

## CEF

- IOS will switch a packet using CEF only if CEF is enabled on the inbound interface (not outbound)
- Composed of two structures: FIB (topology-driven 8-8-8-8 mtrie) and adjacency table where recursive next-hops are automaticaly and immediately resolved
- Cache building is not triggered by first packet but for all entries in a routing table. All changes in routing table are automaticaly reflected in FIB
- ***(G) ip cef [distributed]***  
  CEF is required for MPLS to work
- (IF) ***ip route-cache cef***  
  Enable CEF on interface if it as been removed

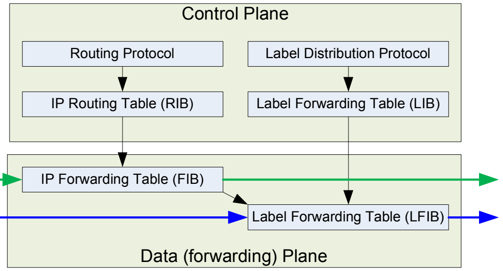
| **Inbound** | **Outbound** | **Method Used** |
|---|---|---|
| CEF | Process | CEF |
| CEF | Fast | CEF |
| Process | CEF | Fast (or process if IPv6) |
| Process | Fast | Fast |
| Fast | CEF | Fast (or process if IPv6) |
| Fast | Process | Process |

### Adjacency Table

- Contains all connected next-hops, interfaces and associated L2 headers
- null – Pointed to Null0
- glean – Destination is attached via broadcast network but MAC is yet unknown
- punt – If CEF is not supported for destination path, switch to next-slower switching
- drop – Cannot be CEF-switched at all. Packets are dropped, but the prefix is checked
- discard – Packets are discarded
- ***show adjacency [detail]***  
  Routes associated with outgoing interface and L2 header

## Load balancing

- ***(IF) ip cef load-sharing {per-packet | per-destination}***  
  Default is per-destination (per flow)
- 16 buckets for hashed destinations (load-sharing is approximate due to small number of buckets)
- ***show ip route <prefix>***  
  If unequal-cost load-balancing is used then for one path more than one hash bucket is used (traffic share count *ratio #*)
- Labels assigned to certain next-hops are inherited by all prefixes using that NH, so the same path is used
- If packet is IPv4 or IPv6 then src-dst pair is used for hashing, otherwise bottom label is used
- Load balancing is possible only if both outgoing paths are labeled or both untagged, no mixing
- ***show ip cef exact-route <src> <dst>***  
  Check which path IPv4 packet will take
- ***show mpls forwarding-table labels <label> exact-path ipv4 <src> <dst>***  
  Displays which path the labeled patcked will take.


Control Plane
Routing Protocol
Label Distribution Protocol
IP Routing Table (RIB)
Label Forwarding Table (LIB)
IPv4 packet
IP Forwarding Table (FIB)
MPLS packet
Label Forwarding Table (LFIB)
Data (forwarding) Plane

## MPLS

### Control Plane

#### LIB

- Every LSR creates local binding of a label-to-an-IPv4-prefix found in FIB. Binding is announced to peers, where they become remote bindings for certain FEC
- From all labels, the downstream router is found in LIB by looking for prefix's next-hop in routing table. This best binding is placed in LFIB
- Label exchange protocols are used to bind labels to FECs
  - RSVP (TE)
  - BGP (VPN)
  - LDP / TDP

### Forwarding Plane

#### LFIB

- Used to forward labeled packets. Populated with the best local and remote labels.
- Received labeled packet is dropped if the label is not in LFIB, even if destination IP exists in FIB
- From all remote bindings the best one is choosen 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 LIB.
- ***show mpls forwarding-table [<ip>] [detail]***  
  Detailed output shows whole label stack, not only pushed label {bottom label, top label}

## IPv4

### Control Plane

#### RIB

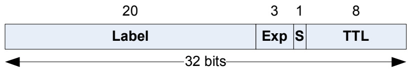
- ***show ip route***  
  Global routing table
- ***show ip route vrf <name>***  
  VRF routing table

### Forwarding Plane

#### FIB

- Contains prefix, automaticaly resolved (recursively) next-hop and L2 adjacency pointer
- attached – Directly reachable via the interface, next-hop is not required
- connected – Directly connected to interface. All connected are attached, but not all attached are connected
- receive – 3 per interface (intf. address + net + br.). Also /32 host addresses
- recursive – Output interface is not directly known via routing protocol from which prefix was received. Recursive lookup is required
- ***show ip cef [vrf <name>] [<ip>] [detail]***  
  CEF is built independently for global routing and each VRF

# MPLS Labels



## Concept

Identifies Forwarding Equivalency Class (FEC) – prefixes belonging to the same path and treated the same way (ex. have the same BGP next-hop). Classification is on ingress LSR

Labels do not have payload information, because intermediate LSRs do not need to know that. Egress LSR knows payload type, as he made the local binding according to the FEC he knows.

Label numbers 0-15 reserved

- 0 – IPv4 explicit Null: Penultimate LSR does not pop the label but sends to egress LSR, which only uses EXP value for QoS and pops the label without LFIB lookup. Only IPv4 lookup is made.
- 1 – router alert v4/v6: Router pops label, examines the packet, performs LFIB lookup and pushes one label. Can be set anywhere except bottom.
- 2 – IPv6 explicit Null
- 3 – IPv4 implicit Null: Advertised to penultimate LSR to pop label and send untagged packet (used for connected and aggregated networks). PHP – **Penultimate Hop Popping** – no need for egress LSR to perform two lookups (label and IP). Only one label is popped off at PHP

***mpls label range <min> <max>***
Default range is 16 – 100000. Use ***show mpls label range*** to verify. Reload is required.

Frame Mode – for protocols with frame-based L2 headers – label inserted between L2 and L3 – **shim header**. Protocol identifier is changed in L2 header to indicate labeled packet

- Eth 0x8847 – IPv4 unicast (0x8848 – IPv4 multicast)
- PPP 0x0281; HDLC 0x8847
- FR 0x80 – IEEE SNAP with Eth 0x8847

Cell Mode – when ATM switch is used as LSR – VPI/VCI used as label because label cannot be instered in every cell

Assignment

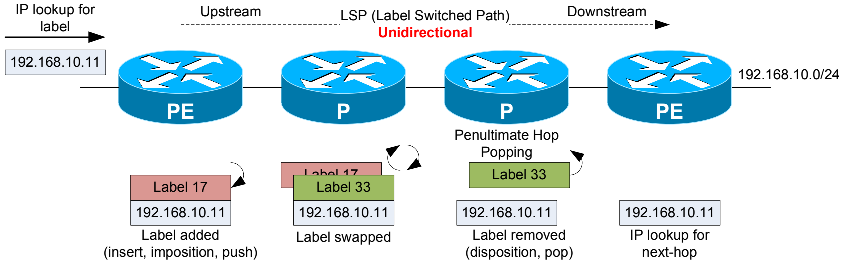
- Locally significant – each LSR binds FEC to label independently (bindings exchanged between LSRs)
- Different labels are assigned for every FEC, except when BGP is used. One label is assigned for all networks with the same BGP next-hop

***debug mpls packet***
Shows interesting label internals {<label> <exp> <ttl>}

## LSP

LSP is unidirectional

Aggregation breaks LSP into separate LSPs. Connectivity may be maintained for plain IPv4, but VPN and TE may be broken.



## 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

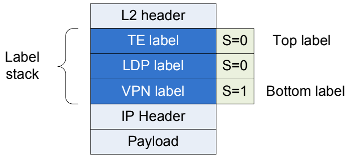
## Retention Modes

- CLR – Conservative: Bindings are removed from LIB after best next-hop is selected and placed in LFIB. Only best binding is stored in LIB – less memory but slow convergence – default for ATM interfaces
- LLR – Liberal
  - Bindings stay in LIB after best next-hop is selected and placed in LFIB
  - Allows faster convergence when link goes down, next best next-hop is selected from LIB
  - Default on any other interfaces (frame mode)

## LSP Control 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 (connected and received from IGP)
  - May cause a packet drop if LSR starts labeling packets and the whole LSP is not set-up yet.
  - Default on any other interfaces (frame mode)

## Label stack



S – bottom of the stack:
1 – bottom label, next is IP header; 0 – more labels follow

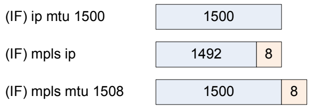
VPN – label identifies VRF, used by PE. Egress LSR does not perform IP lookup for VPN label, because LFIB already points to proper next-hop along with interface and L2 rewrite data

LDP – used by P routers to label-switch packets between LSRs

TE – identified TE tunnel endpoint, used by P, and PE routers

# MPLS MTU/TTL

## MTU



***(IF) mpls mtu 1512***
Defines how large a labeled packet can be. Recommended 1512 for 3 labels (baby giant). The ***ip mtu*** defines how large L3 packet can be when sending on L2 link.

When MPLS is enabled on LAN interface, MPLS MTU is automaticaly increased when labeled packet is to be sent. But, on WAN interfaces MPLS MTU stays the same as IP MTU, so in fact IP MTU is decreased (fragmentation)

MPLS MTU must be set properly on both sides of the link. Interface with lower MTU will receive larger packet, bot it will not send larger packet to the interface (depending on the side with too low MTU, the „ICMP Fragmentation Needed andDF set" may, or may not be received by the source.

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.

MPLS MTU is by default the same as interface MTU. If interface MTU is changed, then MPLS MTU is also automaticaly changed to the same value, but if MPLT MTU is manualy changed, then IP MTU stays the same.

All devices along the L2 path must support baby giant frames

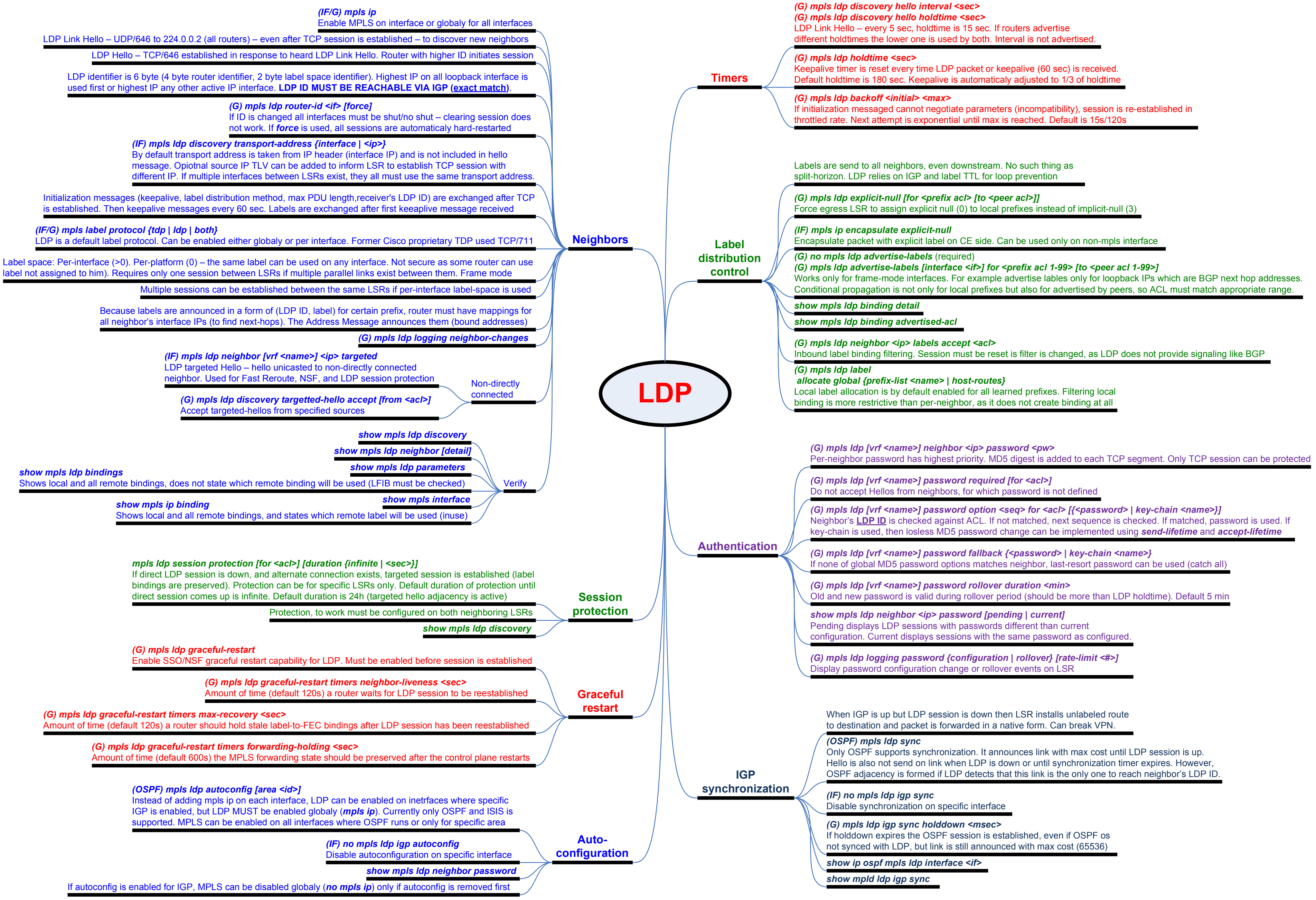
***show mpls interface <if> detail***

## TTL

TTL propagation is enabled by default. If MPLS TTL is higher than IP TTL on egress router then IP TTL is overwritten with label TTL, otherwise it is not ( loop prevention)

***(G) no mpls ip propagate-ttl [forwarded | local]***
Disable TTL propagation for forwarded or localy generated or both types of packets. If propagation is disabled, label TTL is set to 255. Egress LSR does not copy label TTL into IP TTL. ISP core is hidden. One hop is shown with cumulated delay.

If TTL reaches zero on P router, ICMP Time Exceeded (with TTL 255) is sent forward along current LSP to destination (downstream) LSR, as P 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 Exceed. AToM packets are dropped, as they contain L2 header behind label.



# LDP

## Neighbors

***(IF/G) mpls ip***
Enable MPLS on interface or globaly for all interfaces

LDP Link Hello – UDP/646 to 224.0.0.2 (all routers) – even after TCP session is established – to discover new neighbors

LDP Hello – TCP/646 established in response to heard LDP Link Hello. Router with higher ID initiates session

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. **LDP ID MUST BE REACHABLE VIA IGP (exact match)**.

***(G) mpls ldp router-id <if> [force]***
If ID is changed all interfaces must be shut/no shut – clearing session does not work. If ***force*** is used, all sessions are automaticaly hard-restarted

***(IF) mpls ldp discovery transport-address {interface | <ip>}***
By default transport address is taken from IP header (interface IP) and is not included in hello message. Opiotnal source IP TLV can be added to inform LSR to establish TCP session with different IP. If multiple interfaces between LSRs exist, they all must use the same transport address.

Initialization messages (keepalive, label distribution method, max PDU length,receiver's LDP ID) are exchanged after TCP is established. Then keepalive messages every 60 sec. Labels are exchanged after first keeaplive message received

***(IF/G) mpls label protocol {tdp | ldp | both}***
LDP is a default label protocol. Can be enabled either globaly or per interface. Former Cisco proprietary TDP used TCP/711

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 LSRs if multiple parallel links exist between them. Frame mode

Multiple sessions can be established between the same LSRs if per-interface label-space is 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)

***(G) mpls ldp logging neighbor-changes***

Non-directly connected

***(IF) mpls ldp neighbor [vrf <name>] <ip> targeted***
LDP targeted Hello – hello unicasted to non-directly connected neighbor. Used for Fast Reroute, NSF, and LDP session protection

***(G) mpls ldp discovery targetted-hello accept [from <acl>]***
Accept targeted-hellos from specified sources

Verify

***show mpls ldp discovery***

***show mpls ldp neighbor [detail]***

***show mpls ldp parameters***

***show mpls ldp bindings***
Shows local and all remote bindings, does not state which remote binding will be used (LFIB must be checked)

***show mpls interface***

***show mpls ip binding***
Shows local and all remote bindings, and states which remote label will be used (inuse)

## Session protection

***mpls ldp session protection [for <acl>] [duration {infinite | <sec>}]***
If direct LDP session is down, and alternate connection exists, targeted session is established (label bindings are preserved). Protection can be for specific LSRs only. Default duration of protection until direct session comes up is infinite. Default duration is 24h (targeted hello adjacency is active)

Protection, to work must be configured on both neighboring LSRs

***show mpls ldp discovery***

## Graceful restart

***(G) mpls ldp graceful-restart***
Enable SSO/NSF graceful restart capability for LDP. Must be enabled before session is established

***(G) mpls ldp graceful-restart timers neighbor-liveness <sec>***
Amount of time (default 120s) a router waits for LDP session to be reestablished

***(G) mpls ldp graceful-restart timers max-recovery <sec>***
Amount of time (default 120s) a router should hold stale label-to-FEC bindings after LDP session has been reestablished

***(G) mpls ldp graceful-restart timers forwarding-holding <sec>***
Amount of time (default 600s) the MPLS forwarding state should be preserved after the control plane restarts

## Auto-configuration

***(OSPF) mpls ldp autoconfig [area <id>]***
Instead of adding mpls ip on each interface, LDP can be enabled on inetrfaces where specific IGP is enabled, but LDP MUST be enabled globaly (***mpls ip***). Currently only OSPF and ISIS is supported. MPLS can be enabled on all interfaces where OSPF runs or only for specific area

***(IF) no mpls ldp igp autoconfig***
Disable autoconfiguration on specific interface

***show mpls ldp neighbor password***

If autoconfig is enabled for IGP, MPLS can be disabled globaly (***no mpls ip***) only if autoconfig is removed first

## Timers

***(G) mpls ldp discovery hello interval <sec>***
***(G) 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.

***(G) mpls ldp holdtime <sec>***
Keepalive timer is reset every time LDP packet or keepalive (60 sec) is received. Default holdtime is 180 sec. Keepalive is automaticaly adjusted to 1/3 of holdtime

***(G) mpls ldp backoff <initial> <max>***
If initialization messaged cannot negotiate parameters (incompatibility), session is re-established in throttled rate. Next attempt is exponential until max is reached. Default is 15s/120s

## Label distribution control

Labels are send to all neighbors, even downstream. No such thing as split-horizon. LDP relies on IGP and label TTL for loop prevention

***(G) mpls ldp explicit-null [for <prefix acl> [to <peer acl>]]***
Force egress LSR to assign explicit null (0) to local prefixes instead of implicit-null (3)

***(IF) mpls ip encapsulate explicit-null***
Encapsulate packet with explicit label on CE side. Can be used only on non-mpls interface

***(G) no mpls ldp advertise-labels*** (required)
***(G) mpls ldp advertise-labels [interface <if>] for <prefix acl 1-99> [to <peer acl 1-99>]***
Works only for frame-mode interfaces. For example advertise lables 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.

***show mpls ldp binding detail***

***show mpls ldp binding advertised-acl***

***(G) mpls ldp neighbor <ip> labels accept <acl>***
Inbound label binding filtering. Session must be reset is filter is changed, as LDP does not provide signaling like BGP

***(G) mpls ldp label***
***allocate global {prefix-list <name> | host-routes}***
Local label allocation is by default enabled for all learned prefixes. Filtering local binding is more restrictive than per-neighbor, as it does not create binding at all

## Authentication

***(G) mpls ldp [vrf <name>] neighbor <ip> password <pw>***
Per-neighbor password has highest priority. MD5 digest is added to each TCP segment. Only TCP session can be protected

***(G) mpls ldp [vrf <name>] password required [for <acl>]***
Do not accept Hellos from neighbors, for which password is not defined

***(G) mpls ldp [vrf <name>] password option <seq> for <acl> [{<password> | key-chain <name>}]***
Neighbor's **LDP ID** is checked against ACL. If not matched, next sequence is checked. If matched, password is used. If key-chain is used, then losless MD5 password change can be implemented using ***send-lifetime*** and ***accept-lifetime***

***(G) mpls ldp [vrf <name>] password fallback {<password> | key-chain <name>}***
If none of global MD5 password options matches neighbor, last-resort password can be used (catch all)

***(G) mpls ldp [vrf <name>] password rollover duration <min>***
Old and new password is valid during rollover period (should be more than LDP holdtime). Default 5 min

***show mpls ldp neighbor <ip> password [pending | current]***
Pending displays LDP sessions with passwords different than current configuration. Current displays sessions with the same password as configured.

***(G) mpls ldp logging password {configuration | rollover} [rate-limit <#>]***
Display password configuration change or rollover events on LSR

## IGP synchronization

When IGP is up but LDP session is down then LSR installs unlabeled route to destination and packet is forwarded in a native form. Can break VPN.

***(OSPF) mpls ldp sync***
Only OSPF supports synchronization. It announces link with max cost until LDP session is up. Hello is also not send on link when 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.

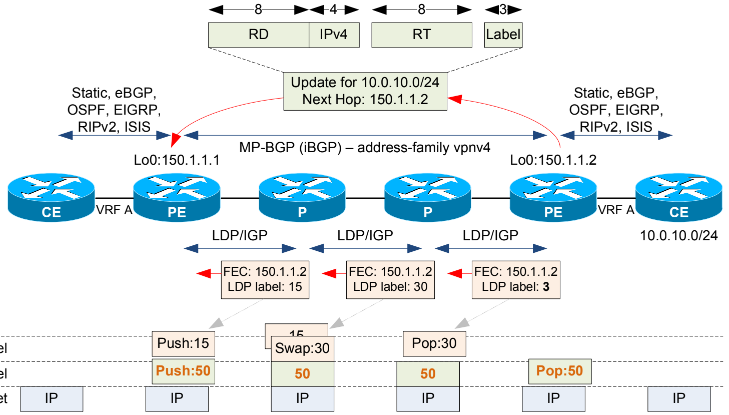
***(IF) no mpls ldp igp sync***
Disable synchronization on specific interface

***(G) mpls ldp igp sync holddown <msec>***
If holddown expires the OSPF session is established, even if OSPF os not synced with LDP, but link is still announced with max cost (65536)

***show ip ospf mpls ldp interface <if>***

***show mpld ldp igp sync***

# L3 VPN



## Concept

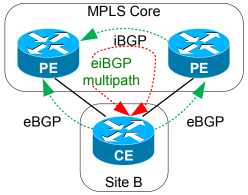
- Legacy
  - Peer-to-peer: IPSec, GRE, L2F, L2TP, PPTP
  - Overlay: FR, ATM VCs. ISP provides L1/L2 (usualy expensive), and does not participate in customer's routing
- MPLS VPN - Collection of sites sharing common routing information
- VPN labels are exchanged between edge LSRs. They describe to which VRF packet will be sent when it reached egress LSR. Intermediate LSRs do not have information abot VPN labels. They only use top label (LDP) to pass traffic
- P routers to not have any knowledge about customer's routes. Only PE routers exchange native routing with customers. P routers only switch labeled packets.
- PE routers exchange routing and label information using BGP (scalable and multi-protocol capability)

## VRF

- *(G) ip vrf <name>*
  Customers' routes must be distinguished on PE routers. Virtual routing and forwarding (VRF) tables are used
- VRF Lite
  - Only VRFs, no MPLS label distribution
  - Lack of scalability. VRFs on separate devices must be connected with separate circuits.
- Route Distinguisher
  - *(VRF) rd <id>*
    64 bit value added to IPv4 address, creating vpnv4 address (96 bits). RD is presened in a form of AS:nn or IP:nn. RD is required for VRF to be operational
  - DOES NOT identify VPN, only provides uniqueness for IP addresses. If CE is multihomed, PEs can use different RD, although they will compose the same VPN
  - VPNv4 addresses are exchanged between PE routers with MP-MGP. When 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 Target
  - Defines VPN membership. Advertised with MP-BGP as extended community.
  - *(VRF) route-target export <RT>*
    Extended RT community is added to all prefixes exported into MP-BGP, regardless of the source protocol
  - *(VRF) route-target import <RT>*
    Route is imported from MP-BGP into VRF only if at least one RT community matches the import RT
  - *(VRF) route-target both <RT>*
    Import and export the same RT. Actualy it is a macro creating the above two entries (import and export)
  - *(VRF) import-map <route-map>*
    Selective import can be used with import map. Route must match both: RT and route-map prefix list, to be imported into VRF
  - *(VRF) export-map <route-map>*
    Export route map can add RT to selected routes. No other action is supported in route-map than ***set extcommunity rt.*** RT is by default overwritten in the prefix, unless ***additive*** keyword is used in route-map
- *(IF) ip vrf forwarding <VRF name>*
  Assign VRF to interface. Existing IP will be REMOVED. Interface can belong to only one VRF
- *(VRF) vpn id <OUI:Index>*
  VPN ID is not used for routing control. It can be used in DHCP server to assign IP per VRF or for RADIUS. OUI is 3 byte hex (like for MAC address manufacturing), Index is 4 byte hex.
- *(VRF) maximum routes <#> {<warn threshold %> | warning-only}*
  Setting limit in VRF is prefered than setting limit in eBGP (CE-PE), which causes session to be reset. To receive warning traps enable ***snmp-server enable traps mpls vpn***
- ***show ip route vrf <name> <prefix>***
- ***show bgp vpnv4 unicast all***
  ***show ip bgp vpnv4 all***
- ***show ip vrf [id]***

## MP-BGP

- Multiprotocol Capabilities
  - Multiprotocol capabilities are exchanged in Open message
  - Introduces MP Reachable NLRI and MP Unreachable NLRI attributes
  - Each attribute has two identifying fileds AFI (2 bytes) and SAFI (1 byte)
    - AFI
      - 0 – reserved
      - 1 – IPv4
      - 2 – IPv6
    - SAFI
      - 1 – unicast
      - 2 – multicast
      - 3 – unicast & mcast
      - 4 – IPv4 label fwding
      - 128 – labeled VPN fwding
- Address Families (same for IPv6)
  - ***address-family vpnv4***
    iBGP prefix and label exchange between PE LSRs
  - ***address-family ipv4 vrf <name>***
    eBGP prefix exchange between PE and CE within a VRF
  - ***address-family ipv4***
    Native BGP sessions for IPv4
- Labels are piggybacked with prefix (AFI 1/SAFI 128) and are composed of 3 bytes – 20 bytes label value (high order bits) and Bottom of the Stack bit (low order bit). Labels are propagated in an opposite direction to data flow
- BGP assignes lables ONLY for prefixes for which it is a next-hop. BGP next-hop cannot be changed across the network (next-hop-self in confederation or inter-AS VPN)
- ***neighbor <ip> activate***
  Neighbors configured in global instance, but activated in specific family
- ***neighbor <ip> send-community {standard | extended | both}***
  Extended communities are automaticaly exchanged if peer is activated. Use ***both*** to also send standard communities.
- ***no bgp default ipv4-unicast***
  If neighbors are already configured in legacy global mode, they can be migrated to address-family-based configuration
- ***show ip bgp vpnv4 all summary***
  Display BGP sessions in all VRF and VPNv4 families
- ***show ip bgp vpnv4 {all | rd <rd> | vrf <vrf>} ...***
  VRF and RD show the same, but on P routers only RD works, as P routers do not have any VPN VRFs
- Multipath
  - Supported only by basic MPLS L3 VPNs (Inter-AS and CSC are not supported). It is configured per-AF
  - ***(BGP) maximum-paths <#>*** - eBGP
  - ***(BGP) maximum-paths ibgp <#> [import <#>]***
    If originating RD is different than egress RD then additionaly we must define how many equal-cust routes can be imported
  - ***(BGP) maximum-paths eibgp <#>*** - eiBGP
  - When CE is multihomed and PEs use RR then multipath may not work, as RR advertises only the best route. The solution is to configure different RDs on both PE, so RR will see two different routes.



- RR
  - ***(BGP) bgp rr-group <ext-comm list>***
    ***(G) ip extcommunity-list <id> {permit | deny} rt <RT>***
    If RR are used they may be impacted by number of routes kept, as they accept all routes (no import scenario as no VRFs are present). RR groups can specify for which RTs the RR should perform route reflection. Configured for vpnv4 AF
- Convergence
  - ***bgp scan-time [import] <5-60 sec>***
    How often MP-BGP prefixes are imported into VRF. Default 60 sec. Newest versions of IOS are event-driven, not based on timers.Also, withdrawn NLRIs are processed immediately, omiting import process to speed up failure recovery

# PE-CE EIGRP

## Features

| | | |
|---|---|---|
| General Metric | 0x8800 – Flags + Tag | |
| | 0x8801 – AS + Delay | |
| | 0x8802 – Reliability + Hop count + BW | |
| | 0x8803 – Reserved + Load + MTU | |
| External | 0x8804 – Remote AS + Remote ID | |
| | 0x8805 – Remote protocol + Remote metric | |

Extended communities are used to describe the route.

If route is internal and AS on both PEs is different then route is redistributed as external.

Down bit (like in OSPF) is not needed, as MP-BGP metric is always 0 so it wins as a direct path

Routes redistributed from MP-BGP into VRF are considered internal, only if remote and local EIGRP AS is the same. Otherwise prefix will be marked as external.

EIGRP topology shows „VPNv4 sourced" prefixes with advertised metric set to zero

## Config

***router eigrp <as>***
***address-family ipv4 vrf <name>***
***autonomous-system <AF AS>***
Only one process is allowed per router so address-family is used for each VRF. Globaly defined AS is used ONLY for native IPv4. You MUST define AS for address-family even if it is the same as global AS

***(BGP) redistribute eigrp <AF AS>***
Configured in address-family, so only routes within proper VRF are redistributed.

***(EIGRP) redistribute bgp <as>***
Metric must be defined either with ***redistribite*** or with ***default-metric*** command. Route will not be redistributed without seed metric defined.

## SOO

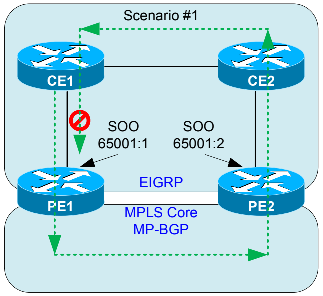
Site of Origin – used for **loop prevention in dual-homed CE** when race condition between EIGRP queries and BGP updates takes place. Attached to VPNv4 route as extended community. EIGRP carries SOO as separate TLV. SOO is added only if it is not already present. If site map matches SOO carried (in any direction) by routing update (via interface where site map is configured) the update is ignored.

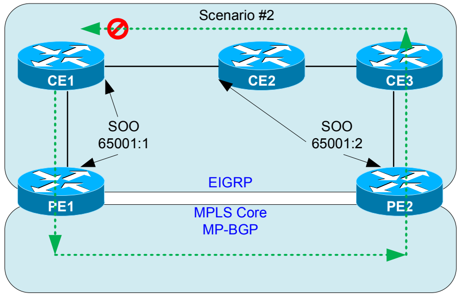
***interface <name>***
***ip vrf forwarding <vrf>***
***ip vrf site-map <route map>***
Adding site map causes EIGRP session reset

***route-map <name> permit <seq>***
***set extcommunity soo <value>***
Configured on PE interface toward CE and between CEs

Each site must be assigned a unique SOO, because if backdoor link between CEs is down, then MPLS core cannot be used as backup for partitioned CE. This solution is slower in convergence, but provides redundancy

To speed up convergence link between CEs can also be marked with SOO, specific for each site. However, if link between CE2 and CE3 is down, MPLS cannot be used to pass traffic between partioned parts of one site





## Cost community

| Cost community Type 0x4301 | POI | ID | Cost |
|---|---|---|---|
| 2B | 1B | 1B | 4B |

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

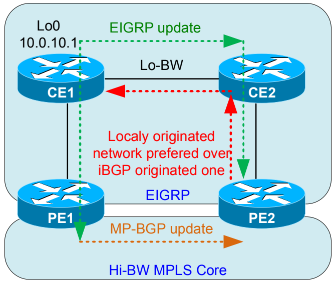
By default localy redistributed prefixed on PE (from CE) have BGP weight set to 32768, so if backdoor link exists, and remote site's prefixes are redistributed by local PE, they are prefered over those received via MP-BGP, even if metric is better via ISP

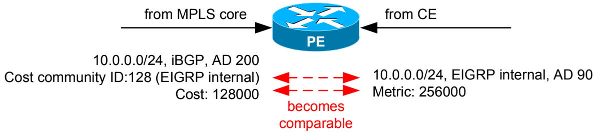
POI (pre-bestpath) existence defines that the cost community should be evaluated before checking if route is localy originated or not (BGP route selection process is modified).

Allows PEs to compare routes coming from EIGRP and iBGP (different ADs). BGP routes carrying cost community can be compared to EIGRP route's metric, becase cost community carries complete composite metric. **Alleviates suboptimal routing over backdoor link**

ID is a tiebreaker when costs are the same. Lower is better.
ID 128 – EIGRP internal routes, 129 – EIGRP external routes

***(BGP) bgp bestpath cost-community ignore***
In certain cases you can disable cost-community





from MPLS core
PE
from CE
10.0.0.0/24, iBGP, AD 200
Cost community ID:128 (EIGRP internal)
Cost: 128000
becomes comparable
10.0.0.0/24, EIGRP internal, AD 90
Metric: 256000

# PE-CE eBGP

## Overlapping CE AS

Each site should have different AS, otherwise, AS path must be manipulated to allow paths with own AS

***(BGP) neighbor <ip> as-override***
Configured on PE for CE peer. When AS-PATH's **last** AS numer (multiple entries can exist if prepending was used) is the same as CE's AS, it is replaced (all instances when prepending was used) with ISP PE's AS

***(BGP) neighbor <ip> allowas-in <1-10>***
Configured on CE for PE peer. CE router will allow own AS in the AS-PATH, but only if it is present no more than # of times

## Config

***address-family ipv4 vrf <name>***
***neighbor <ip> remote-as <as>***
***neighbor <ip> activate***
CE neighbors are configured in VRF address family

Redistribution from eBGP into MP-MGP is automatic

## SOO

Overriding AS caues route to be injected back to multihomed CE. SOO can be used to prevent loops. SOO has the same meaning as in EIGRP, so the same scenarios can be used to use MPLS core as backup in case backdoor link is down.

***(BGP) neighbor <ip> soo <value>***
Method #1. Configured on PE for CE neighbor. Automaticaly sets SOO for inbound and outbound prefixes

***(BGP) neighbor <ip> route-map <name> in***
Method #2. Configured on PE for CE neighbor. Route map sets SOO ext community for incoming prefixes

# PE-CE Other

## Static

***(G) ip route vrf <name> <net> <mask> {<gw> | <interface>}***
You can use any interface (different VRF of native) as long as it is p2p interface

***(BGP) redistribute static***

***(G) ip route static inter-vrf***
Enabled by default. Allows static routes in global config (or other VRF) to point into interface in different VRF. If disabled, allows avoiding interface name typos when adding customer's static routes.

## RIPv2

***router rip***
***address-family ipv4 vrf <name>***
Only one process is allowed per router so address-family is used for each VRF

***(RIP) redistribute bgp <as> metric {<hop> | transparent}***
When RIP is redistributed on peer LSR into BGP, hop count is coppied into MED. If ***transparent*** metric is used, hop count is derived back from MED. Default metric can be also defined with ***default-metric <hop>***

***(BGP) redistribute rip***

There is no mechanizm to set preference for MP-BGP routes if backdoor link is used.

## Internet access

### Static default

***(G) ip route vrf <name> 0.0.0.0 0.0.0.0 <NH> global***
Default route for all sites within VPN (should be redistributed into MP-BGP). Global keyword means that next-hop should be reselved from global native routing table, even though the route itself is within the VRF

***(G) ip route <net> <mask> <CE interface>***
Static route in global table for cusomter's public IPs pointed into interface toward CE (for returning traffic)

Other solutions are: seprate PE-CE circuit for native internet access with full BGP feed (native ipv4 BGP peering), extranet vith Internet VRF or VRF-aware NAT

# PE-CE OSPF

## Features

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

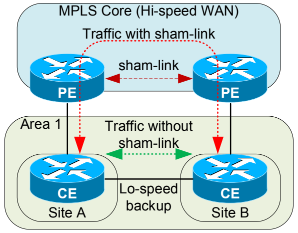
Information about route is propagated using extended community called RT (route type, different than route target), OSPF router ID (4 bytes), and OSPF domain (process number) ID (2 bytes)

***RT:<area 4Bytes>:<route type 1Byte>:<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

***(OSPF) domain-id <id>***
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 then 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 routes is coppied into MED. MED can be manipulated manualy to influence path selection

## Sham Link



Intra-area route is prefered than inter-area. If backup link exists between sites it will be prefered no matter what cost inter-area routes have. Also OSPF has lower AD (110) than iBGP (200)

Sham link is an intra-area unnumbered p2p control link carried over superbackbone (in the same area as PEs). It's a demand circuit so no periodic hellos are sent, and LSAs do not age out

OSPF adjacency is established. LSAs are exchanged, but they are used only for path caluclations. 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

***(OSPF) area <id>sham-link <src IP> <dst IP> [cost <cost>]***
Cost should be set to lower value so it is prefered over backdoor link.

***(BGP) network </32 loopback> mask 255.255.255.255***

***show ip ospf sham-link***

## Config

***(G) router ospf <id> vrf <name>***
Multiple OSPF instances can exist, so process is configured per VRF

***(OSPF) redistribute bgp <as> subnets***

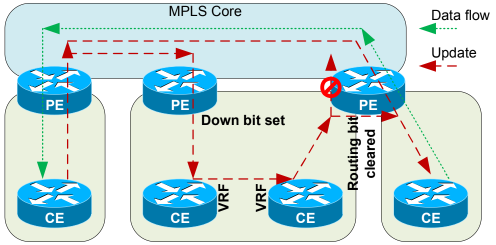
***(BGP) redistribute ospf <id> match {internal | external 1 | external 2}***
If match is not defined only internal routes are redistributed.

## Domain tag

***(OSPF) domain-tag <value>***
When external routes are redistributed from MP-BGP into OSPF the OSPF tag is set to BGP AS. Tag is propagated within OSPF domain, even between different processes (where down-bit is cleared). PE route will not redistribute OSPF route to MP-BGP if tag matches BGP AS (loop prevention)

***(OSPF) redistribute bgp <as> subnets tag <tag>***

## Down Bit



Dual-homed area loop prevention

Automaticaly set in LSA 3 (only) header options field when routes are redistributed from MP-BGP into OSPF. When down bit is set for prefix received on interface which is configured with VRF, the OSPF will never use this LSA for SPF calculations. PE will not redistribute such routes back to MP-BGP

When down bit is set, routing bit gets cleared on PE. Route will not be placed into routing table even if it is the best path. Otherwise sub-optimal routing would take place (through transiting area, not mpls superbackbone)

***(OSPF) capability vrf-lite***
Required on CEs if VRF Lite 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 LSA5, which does not fall under this loop-prevention solution