Cisco Routing and Swithing Quick Review Kit

By: Krzysztof Załęski
CCIE R&S #24081

ver. 20151025

This Booklet is dedicated to my wife and my
kids, for their patience and understanding

Copyright information

Cisco Routing and Switching Quick Review Kit
By: Krzysztof Załęski, CCIE R&S #24081
http://ccie24081.wordpress.com
cshyshtof@gmail.com

ver. 20151025

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. The main goal of this material is to provide quick and easy-to-skim method of refreshing one'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, Cisco Press books, and other well-known vendors' 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.

Table of Contents

(#) – enable command
(G) – global command
(IF) – interface command
(RM) – route-map command
(CM) – class-map command
(PM) – policy-map command
… you get the idea...

1B	1B	2B		2B
Address 0xFF	Control 0x03	Protocol	Data	FCS

PPP

Features

LCP – to establish, configure, and test the data link connection – mandatory phase, must be in OPEN phase to proceed with NCP and authentication

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

(G) no peer neighbor route
Peers' IP addresses are send in IPCP negotiation and they show up as /32 connected networks in addition to /30 subnets. Host routes received from peer can be discarded with this command.

Users must be defined with ***password*** keyword, the ***secret*** is not supported (bidir decryption)

Address IP can be sent to peer (like DHCP). Such address is always seen as /32 host route

RTA:
(IF) ip address negotiated

RTB (option A):
(IF) peer default ip address <remote ip>

RTB (option B):
(G) ip adress-pool local
(G) ip local pool <name> <first IP> <last IP>
(IF) peer default ip address pool <name>

CHAP

CHAP is a 3-way handshake authentication method based on challenge-response. No clear-text passwords are sent across the link

Done upon initial link establishment and may be repeated any time after the link has been established

(IF) ppp authentication chap
Router with this command requests the otreh side to authenticate with CHAP

(IF) ppp chap hostname <name>
Send alternate hostname as a challenge. By default, real hostname is sent as username

(IF) ppp chap password <pass>
This password is used if global username is not configured

(IF) 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)

(IF) 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

(IF) ppp chap wait
The router will not authenticate to a peer that requests CHAP authentication until the peer has authenticated itself to the router

CHAP will fail if hostnames are the same on both sides

MSCHAP and EAP are also supported

LFI

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

(IF) ppp multilink fragment-delay <msec> - Configured on a single physical interface

(IF) ppp multilink interleave

PAP

PAP (Password Authentication Protocol) is a 2-way authentication method, sending clear-text login and password (request-response). Can be uni- or b-directional

(IF) ppp authentication pap
Router with this command requests other side to authenticate with PAP

(IF) ppp pap sent-username <username> password <password>
Send hostname and a password in response to PAP request

(IF) ppp pap wait
The router will not authenticate to a peer that requests PAP authentication until the peer has authenticated itself to the router (bi-directional authentication configuration required)

(IF) 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

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

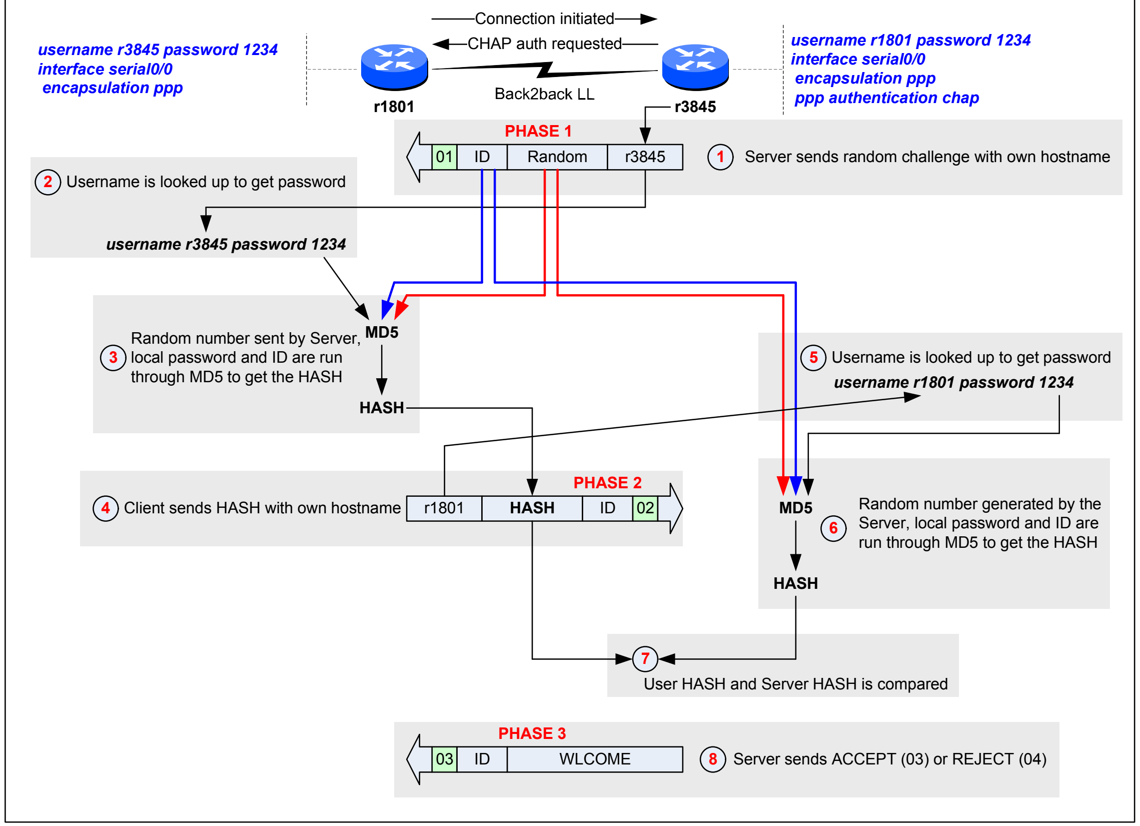
Client:

```
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 authen. 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 user R1
```

CHAP Unidirectional 3-way challenge



r1801:

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

r3845:

```
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

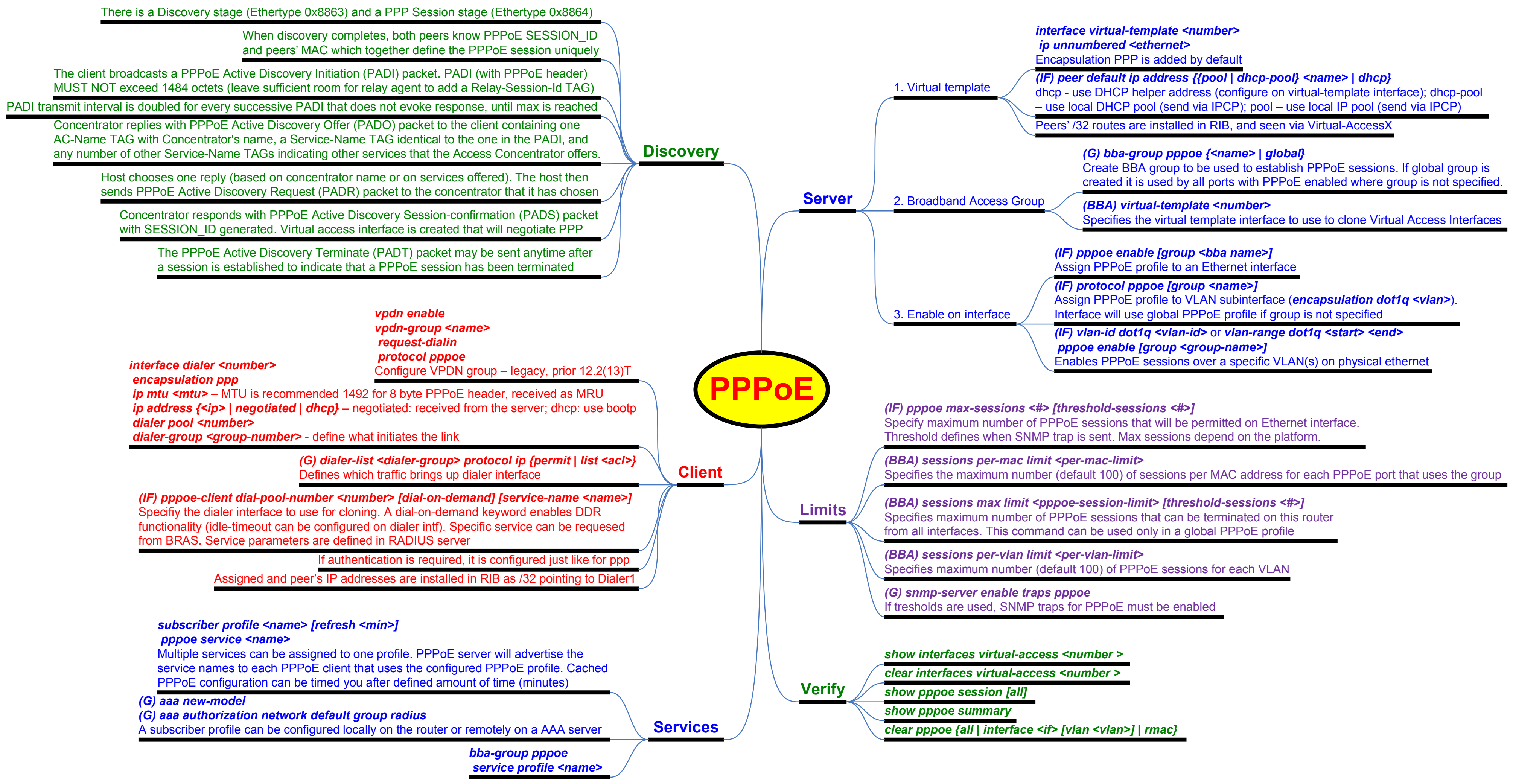
4 Client sends HASH with own hostname

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

Discovery

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

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> – MTU is recommended 1492 for 8 byte PPPoE header, received as MRU
ip address {<ip> | negotiated | dhcp} – negotiated: received from the server; dhcp: use bootp
dialer pool <number>
dialer-group <group-number> - define what initiates the link

(G) 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 requesed from BRAS. Service parameters are defined in RADIUS server

If authentication is required, it is configured just like for ppp

Assigned and peer's IP addresses are installed in RIB as /32 pointing to Dialer1

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)

(G) aaa new-model
(G) 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>

Server

1. Virtual template

interface virtual-template <number>
ip unnumbered <ethernet>
Encapsulation PPP is added by default

(IF) peer default ip address {{pool | dhcp-pool} <name> | dhcp}
dhcp - use DHCP helper address (configure on virtual-template interface); dhcp-pool – use local DHCP pool (send via IPCP); pool – use local IP pool (send via IPCP)

Peers' /32 routes are installed in RIB, and seen via Virtual-AccessX

2. Broadband Access Group

(G) 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 <bba name>]
Assign PPPoE profile to an Ethernet interface

(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(s) on physical ethernet

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

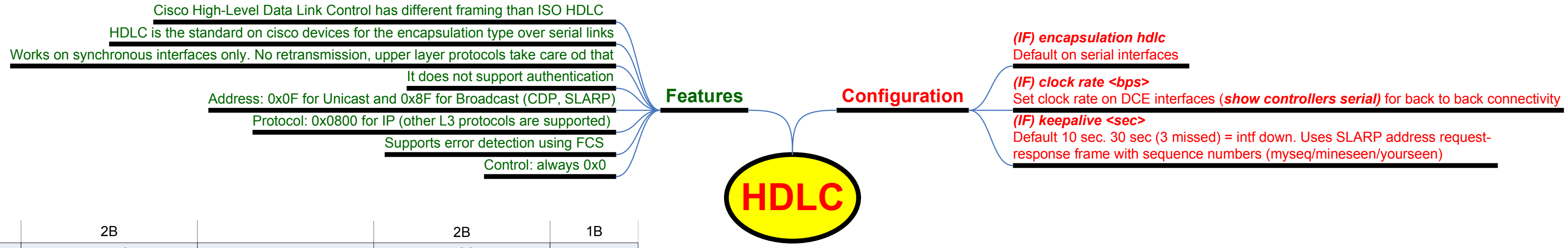
(BBA) sessions max limit <pppoe-session-limit> [threshold-sessions <#>]
Specifies maximum number of PPPoE sessions that can be terminated on this router from all interfaces. This command can be used only in a global PPPoE profile

(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}



HDLC

Features

Cisco High-Level Data Link Control has different framing than ISO HDLC

HDLC is the standard on cisco devices for the encapsulation type over serial links

Works on synchronous interfaces only. No retransmission, upper layer protocols take care od that

It does not support authentication

Address: 0x0F for Unicast and 0x8F for Broadcast (CDP, SLARP)

Protocol: 0x0800 for IP (other L3 protocols are supported)

Supports error detection using FCS

Control: always 0x0

Configuration

(IF) encapsulation hdlc
Default on serial interfaces

(IF) clock rate <bps>
Set clock rate on DCE interfaces (***show controllers serial)*** for back to back connectivity

(IF) keepalive <sec>
Default 10 sec. 30 sec (3 missed) = intf down. Uses SLARP address request-response frame with sequence numbers (myseq/mineseen/yourseen)

1B	1B	2B		2B	1B
Address	Control	Protocol Code	Data	FCS	Flag

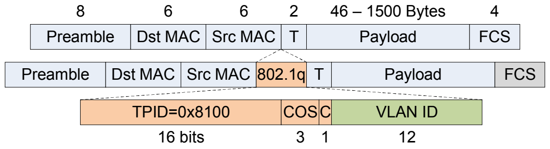
VLAN

Types

- Reserved: **0, 1002 – 1005, 4095**
- Normal range **1-1001**
  - Can be configured in Server and Transparent modes
  - VLAN1 cannot be deleted, and it's name (default) cannot be changed
  - Propagated by VTP
- Extended range **1006 - 4094**
  - Supported only in Transparent and VTP v3 modes. Not propagated by VTP v1 and v2, but propagated by v3
  - Not supported in VLAN database configuration mode (***vlan database***)
  - ***(G) vlan internal allocation policy {ascending | descending}***
    Each routed port on a Catalyst 3550 switch creates an internal VLAN for its use. These internal VLANs use extended-range VLAN numbers, and such 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***)
  - Extended VLANs cannot be pruned
- Native
  - By default VLAN1 is native on all trunks (untagged frames are assigned to native VLAN)
  - Not supported on ISL trunks – all frames are tagged
  - ***(G) vlan dot1q tag native***
    Emulates ISL behaviour on 802.1q trunks for tagging native VLAN (required for QinQ). The switch accepts untagged packets, but sends only tagged packets.
  - ***(IF) encapsulation dot1q <vlan-id> native***
    By default, native VLAN is terminated on physical router interface. It can be processed by a subinterface i ***native*** keyword is used
  - ***(IF) switchport trunk native vlan <id>***
    Native VLAN, even though it is not tagged, it MUST be allowed with ***switchport trunk allowed vlan*** command if it is used
  - CDP can detect misconfigured native VLANs – VLAN hopping!
- Voice
  - The Port Fast feature is automatically enabled when voice VLAN is configured
  - 802.1q
    - ***(IF) switchport voice vlan <id>***
      If port is configured as access, the switch will convert it internally into a trunk
    - VLAN number is communicated to phone via CDPv2 (required for IPPhones)
  - 802.1p
    - ***(IF) switchport voice vlan dot1p***
      VLAN 0 is used to carry voice traffic
    - Switch treats frames with 802.1q tag set to 0 as it was an access port, but honors 802.1p COS field for QoS. Traffic is then assigned back to native VLAN
  - untagged
    - ***(IF) switchport voice vlan untagged***
  - none
    - ***(IF) switchport voice vlan none***
      Allow the phone to use its own configuration to send untagged voice traffic

Trunking

- DTP
  - Switches must be in the same VTP domain. Default mode is Desirable on 3550 only. It is Auto on 3560
  - Routers do NOT understand DTP protocol. Trunk must be staticaly defined on switch port
  - Messages sent every 30 sec (300sec timeout) to 01-00-0C-CC-CC-CC (ISL – VLAN1, 802.1q – Native)
  - If both switches support ISL and 802.1q then ISL has priority
  - ***(IF) switchport mode trunk*** – always trunk, sends DTP to the other side
  - ***(IF) switchport mode access*** – always access, DTP is disabled
  - ***(IF) switchport mode dynamic desirable*** – sends negotiation DTP messages
  - ***(IF) switchport mode dynamic auto*** – replies to negotiation DTP messages
  - ***(IF) switchport nonegotiate***
    Disable sending of DTP messages. Can be used only if static trunking is configured
  - If DTP does not netogiate trunk, port becomes access assigned to VLAN (default 1)
  - ***show interface [<if>] trunk***
- ISL
  - Cisco proprietary protocol supporting up to 1024 VLANs - depreciated
  - 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
  - Canonical Format Indicator (CFI) is used only for TokenRing frames
  - TPID is in the same place as previous EtherType (T) field, indicating the frame is tagged. Real EtherType follows 802.1q tag



VMPS

- ***(IF) switchport access vlan dynamic***
  Switch (client) starts talking to server using VLAN Query Protocol (VQP)
- When server configured in secure mode the port is shutdown if MAC-to-VLAN mapping is not in database. In open mode, access is denied but port stays up
- ***(G) vmps server <ip> [primary]***
- ***(G) vmps reconfirm <sec>*** - default refresh is every 60 min
- ***(G) vmps retry <#>*** - default 3 times
- ***show vmps***

Private VLAN

Features

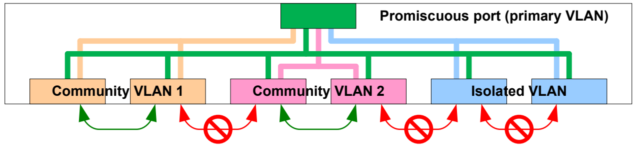
- All hosts can be in the same subnet. VTP transparent is required (unless VTP v.3 is used)
- When you enable DHCP snooping on primary VLAN, it is propagated to the secondary VLANs
- STP runs only on primary VLAN. Community and isolated VLANs do not have STP instance
- Configure private VLANs on all intermediate devices, including devices that have no private-VLAN ports
- Prevent any communication at Layer 2, however hosts can communicate with each other at Layer 3
- ***show vlan private-vlan***

Primary

- All devices can access it. Isolated and community VLANs must be associated with primary VLAN
- L3 devices communicate with a private VLAN only through the primary VLAN and not through secondary VLANs, so on L3 switch configure SVIs only for primary VLANs
- Any configuration on the primary VLAN is propagated to the secondary VLAN SVIs
- ***vlan <id>***
  private-vlan primary
  private-vlan association <list>
- ***interface <if>***
  switchport mode private-vlan promiscuous
  switchport private-vlan mapping <pri> <list>
  Define L2 trunk as primary with secondary VLANs
  - ***interface vlan <id>***
    private-vlan mapping <list>
    Define SVI port as primary

Secondary

- Dynamic MAC addresses learned in private VLANs are replicated in the primary VLAN
- **community VLAN**
  Can talk to Primary and to each other within a community VLAN, but not to other community VLANs. There can be many community VLANs
  - ***(VLAN) private-vlan community***
    Define VLAN as community
- **isolated VLAN**
  Can talk only to Primary. Only one isolated VLAN
  - ***(VLAN) private-vlan isolated***
    Define VLAN as isolated
- ***(IF) switchport mode private-vlan host***
  Define L2 port as secondary VLAN
- ***(IF) switchport private-vlan host-association <pri> <sec>***
  Assign L2 port to community or isolated VLAN



VTP

Features

- Works only over trunk ports. Uses MAC: 01:00:0C:CC:CC:CC and LLC SNAP SSNAP:AA, DSNAP:AA. SNAP header type: 2003
- By default, VTP operates in version 1. All switches must use the same version
- Configuration revision is 32 bits, it is incremented by 1 on every change. To reset revision number, change mode to transparent or domain name
- Supports only basic VLANs (2-1001)
- ***(G) 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)

Domain

- ***(G) vtp domain <name>***
  Initialy a switch is in VTP no-management-domain (NULL) state until it receives an advertisement for a domain or domain is configured. Domain is 0-padded to 32 bytes
- If no domain is configured, 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
- DTP sends VTP domain in negotiation messages. If domains are different, trunk will not come up. Static trunk must be configured then

Messages

- **Summary advertisement** - sent every 5 min, and on every change. Contains domain name, revision, updater id (IP), timestamp, md5 digest and followers (set if adv is due to change, it means Subset Advertisements will follow)
- **Subset advertisement** - contains VLANs (status, vlan type, isl vlan id, mtu size, 802.10 index, vlan name - padded to multiples of 4 bytes). VLANs are sent in ordered form (lower vlans first)
- **Advertisement request** - sent when switch is reset, domain has been changed, or summary advertisement with higher revision was received

Verify

- ***show vtp status***
- ***show vtp password***
- ***show vtp counters***

Modes

- Server
  - Can add, delete and modify VLANs. Propagates changes through domain. Accepts messages from the same domain
  - Does not propagate info untill domain is configured
  - Information is stored only in vlan.dat file on flash:
  - ***(G) vtp mode server***
- Client
  - Accepts VTP messages within domain. No modifications allowed
  - ***(G) vtp mode client***
- Transparent
  - Can add, delete and modify VLANs. Does NOT propagate anything, nor accepts any VTP messages. Required is extended VLANs need to be configured, as well as Private VLANs
  - Can forward VTP messages only in VTP ver 2
  - If transparent is between clients and servers, you still need to manualy configure VLANs on transparent, otherwise traffic for unconfigured VLANs will be dropped
  - ***(G) vtp mode transparent***
  - Revision is always set to 0

Pruning

- ***(G) vtp pruning***
  Enabling VTP pruning on a VTP server enables pruning for the entire domain
- Transparent switches do not participate in pruning, as they do not analyze VTP payload
- ***(IF) switchport trunk prunning vlan <list>***
  VTP pruning blocks unneeded, flooded traffic (unknown unicast, broadcast) within VLANs (on trunk ports) that are included in the pruning-eligible list. Only VLANs 2-1001 are pruning eligible
- ***(IF) switchport trunk allowed vlan <list>***
  Only listed VLANs are allowed to pass the trunk port, but all are announced via VTP on that port. It can be used as a pruning mechanism on Transparent switches. When you remove VLAN 1 from a trunk port, the interface still continues to send and receive management traffic (CDP, PAgP, LACP, DTP, VTP) within VLAN 1. STP still runs for pruned VLANs
- ***show interface <if> prunning***

Security

- ***(G) vtp password <pw>***
  Password can be revealed with command ***show vtp password***

VTPv3

Features

- Supports whole range of VLANs (2 – 4095), so "***spanning-tree extended system-id***" MUST be set
- Supports propagation of Private VLANs. Supports other databases, not only VLANs (MST mappings)
- If switch is not in MST mode, but receives the MST mapping update from primary server, it still stores it localy. It will be instantly used when MST is enabled
- Provides protection from database override caused by adding new switch to the network with higher revision – only primary server can update other switches
- Domain is not learned from first announcement heard, (if it is set to NULL on the switch). To configure v3, domain MUST be set manually
- ***(G) vtp version 3***
  Ver.3 is compatible with Ver.2 on per-port basis, but NOT with ver.1. If switch discovers v2 messages it will send BOTH v3 and v2 messages on that interface as long as v2 is heard. However, v3 switch cannot be updated by v2 switch
- Advertisements include primary server ID, so sanity check can be performed

Security

- ***(G) vtp password <pw> [hidden | secret]***
- If ***hidden*** password is defined, it cannot be revealed with show command anymore (hash is displayed)
- Secret keyword allows to configured hashed password directly (must be 32 hex numbers)
- To promote secondary server to primary role, you will be asked for password if hidden option is used

Verify

- ***show vtp devices [conflict]***
- ***show vtp interface [<if>]***

Roles

- Client
  - If MST is used, after booting all VLANs are assigned to default IST until VTP v.3 message arrives. Client stores VLANs in RAM only
- Server
  - **Primary** and **secondary** server. Servers store VLANs on RAM, and NVRAM. VLANs can be configured only on primary server (regardless or revision number). Secondary is just for backing up configurations
  - ***(G) vtp primary [vlan | stp]***
    Only one server in a whole domain can be promoted as primary server. There can be two separate devices, each with different role (per instance: VLAN, MST)
  - Default role for VLAN instance is secondary Server. Other instances (MST) will be Transparent
  - Former primary server, after reload, will be reverted back to secondary server
- Off
  - ***(IF) no vtp***
    If disabled on interface, all instances (VLAN, MST) become disabled. Works only on trunk ports
  - ***(G) vtp mode off [vlan | mst]***
    Disables VTP on all trunk interfaces. However, only specific instance (VLAN, MST) can be disabled
  - Acts like transparent mode, but DOES NOT relay any messages
- Transparent
  - Just like v.2

	Relay	Configure	Save
Primary server	Y	Y	Y
Secondary server	Y	No	Y
Client	Y	No	No
Transparent	Y	Y	Y
Off	No	Y	Y

Pruning

- ***(G) vtp pruning***
  In VTP v.3 you have to enable pruning manually in every switch of the domain
- Reserved and extended VLANs still cannot be pruned

PVST+

Timers & Features

- Passive protocols, slow convergence, lots of waiting for timeouts
- Based on IEEE 802.1D standard and includes Cisco proprietary extensions such as BackboneFast, UplinkFast, and PortFast. PVST was supported only on ISL trunks
- ***(G) spanning-tree vlan <id> hello-time <sec>***
  BPDU generation (default is 2 sec). Skew detection sends syslog if switch detects delay in BPDU arrival (non-root). Syslog is rate-limited 1msg/60sec, unless delay is MaxAge/2 (10 sec), then shown immediately
- ***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
- Blocking (20sec) => Listening (15sec) => Learning (15 sec) => Forwarding
- Changing the STP protocol always makes the tree to rebuild (ports go through all stages)
- Blocking
  - Discards frames received on the interface. Discards frames switched from another interface for forwarding. Does not learn addresses. Receives BPDUs
- Listening
  - Discards frames received on the interface. Discards frames switched from another interface for forwarding. Does not learn addresses. Receives BPDUs, learns topology
- Learning
  - Discards frames received on the interface. Discards frames switched from another interface for forwarding. Learns MAC addresses. Receives BPDUs
- Bridges are not interested in local timers, they use timers send by Root Hellos.
- Each BPDU sent by root, contains the Age timer. Root sets age to zero, every other switch adds 1 sec (transit delay), so BPDU shows how many hops away the root is
- The max-age timer is reset on every BPDU receipt. This timer does not count down, but the counter starts from Age timer, and when it reaches max-age, BPDU is aged out. So, the further the switch, the less time is left for max-age. Ex. first switch from the root has 20 sec, second switch has 19 sec to age out BPDU...

1. Elect the Root bridge

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

- **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 (always – no way to find current secondary's priority)
  - ***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 on interface per VLAN
  - Forwarder's ID (Bridge ID of the switch that forwarded BPDU)
  - Forwarder's port priority – configured on interface out of which BPDU is sent
  - Forwarder's port number – outgoing interface

2. Determine the Root Port

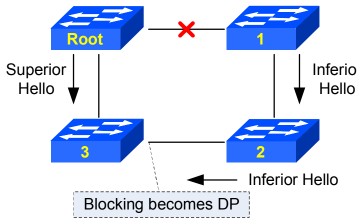
- **1.** Port on which Hello was received with lowest Cost (after adding own cost)
  (IF) spanning-tree vlan <id> cost <path-cost> (configured on root port)
- **2.** Lowest forwarder's Bridge ID – the one who sent BPDU to us
- **3.** Lowest forwarder's port priority (default 128, in increments of 16)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

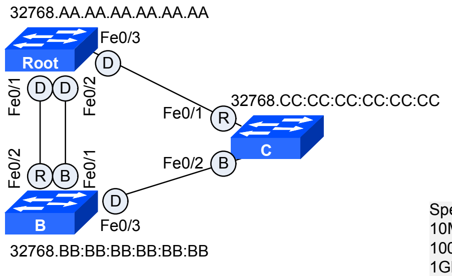
3. Determine Designated Ports

- Only one switch can forward traffic to the same segment
- BPDUs forwarded with lowest advertised cost (without adding own cost) define DP
- Switch with inferior BPDU stops forwarding them to the segment
- If advertised costs are the same the tiebreaker is exactly the same as for Root Port

4. Topology change

- Switches receive BPDUs on all ports, even blocked ports. They store and relay only best BPDU (from root). If superior is heard, previous is discarded, and new one is stored and relayed.
- If 10 Hellos are missed (Maxage 20 sec) the switch thinks it is a root and starts sending own Hellos again
- Any change resulted in port to be unblocked, forces that port to go through Listening and Learning (30 sec)
- If a switch receives new, different „best" Hello on blocking port, and it still hears superior Hello on different port, it switches over the first port from blocking to DP and starts forwarding superior Hellos
- Switch ignores worse BPDUs untill max-age timer expires, even if his own BPDU is to be the best (in case current path to root is lost, and switch tries to declare itself as a root - only if there are no other potential ports receiving superior BPDU from current root, so the port transitions to listening and learning, otherwise, switch generates own BPDUs thinking it is a root)
- All switches need to be informed about the change to timeout CAM
  - Switch sends TCN BPDU every hello time (localy defined, not from root), on root port toward Root every until ACKed by upstream switch
  - Upstream switch ACKs with next BPDU, setting Topology Change Ack (TCA) bit, and sends TC upward, until root is reached
  - When root receives TCN, it sets TCA for next BPDUs so all switches are notified
  - All switches use Forward Delay Timeout (15 sec) to timeout CAM (default is 300 sec) for period of MaxAge + ForwardDelay (35 sec). Root sets TC in Hellos for the period of that time
  - It's better than clearing MAC table, as there might be hosts successfuly communicating with each other





Speed	802.1d	RSTP
10Mb/s	100	2.000.000
100Mb/s	19	200.000
1Gb/s	4	20.000
2Gb/s	3	10.000
3-7Gb/s	2	
8Gb/s	1	
10Gb/2	1	2.000
20-40Gb/s	1	

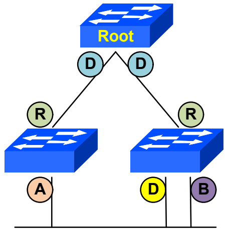
RSTP 802.1w

Features

- *(G) spanning-tree mode rapid-pvst*
- BPDUs are sent to 01:80:C2:00:00:00
- BPDU ver.2 is used (unused fields are now used to define port role, port state, and proposal and agreement states - 802.1d used only two bits: TC and TCAck)
- RSTP decouples the role and the state of port. 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 2 sec
- Maxage only 3 Hello misses (fast aging). Basicaly RSTP is not timer-based
- 802.1w is compatible with 802.1d. Port working as RTSP, when it comes up, starts a migration timer for 3 seconds. If port receives 802.1d BPDU, it transitions to 802.1d. When legacy switch is removed, RSTP switch continues working as 802.1d. Manual restart is required on that port.
- RSTP is able to actively confirm that port can safely transit to forwarding state without relying on any timers. Switch relies now on two variables: edge port and link type
- Now implemented in 802.1D-2004

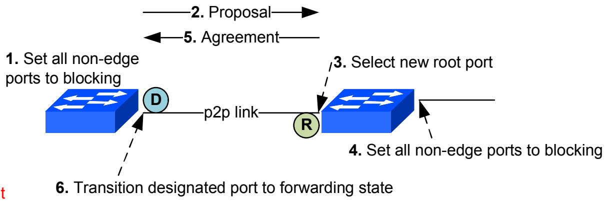
Port roles

- New port roles used for fast convergence
  - **B**ackup port – Receives better BPDU from the same switch on the segment. Provides redundant path to the same segment. Usually does not guarantee a redundant path to root, but can be also Alternate port if no other Alternate ports are available
  - **A**lternate port – Receives better BPDU from the other switch on one segment. Provides redundant path to the root. There can be Alt ports on one switch
- Port types
  - **point-to-point**
    - Full duplex port (only two switches on LAN segment) – simple and fast sync process
    - Required for sync process with another switch, otherwise legacy STP negotiation
    - ***(IF) spanning-tree link-type point-to-point***
The p2p state can be manualy forced if HDX (half-duplex) is used
    - ***show spanning tree vlan <#>***
P2p – RSTP neighbor; P2p Peer(STP) – legacy neighbor
  - **shared**
    - Ports with Half Duplex require arbitration, slow and complicated sync process. Does not support RSTP and STP interoperation.
  - **edge**
    - ***(IF) spanning-tree portfast [trunk]***
Highly recommended on all edge ports



Convergence

- Sync
  - If root port changes or better root information is received, the bridge sends a proposal only out of all downstream 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. This is agreement, only if this switch does not have better root information
  - Sync stops when there is no more leafs, or Reject is received (downstream switch has better root information)
  - If designated discarding port does not receive agreement (downstream does not understand RSTP or is blocking), port slowly transitions for forwarding like 802.1d
  - Proposals are ignored on blocked ports, unless inferior BPDU is received. If local root info is better, switch immediately sends back proposal so inferior switch can quickly adapt. If local info is worse, new sync process begins.
- Topology change
  - Only link-up causes TC, as new path may be build. If link goes down, simple sync proces takes place. Edge ports do not generate TCN, nor sync, regardless of their state change (up or down)
  - 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 (except the receiving one) until the TC timer expires (2x hello). Process contingues through whole domain
  - TCNs are never flooded to edge ports, as there are no switches there
  - Due to MAC flushing, excessive unknown unicast flooding takes place
  - If alternate port is present, sync is dome on that port and fast reconvergence is performed
  - If no alternate port is availabe, declare itself as a root and perform global sync


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

BPDU Frame

TCN BPDU
Type value: 128

BPDU Frame
Protocol ID (2B)
Protocol Version ID (1B)
BPDU Type (1B)
Flags (1B)
Root ID (8B)
Root Path Cost (4B)
Bridge ID (8B)
Port ID (2B)
Message Age (2B)
Max Age (2B)
Hello Time (2B)
Forward Delay (2B)

BPDU Flags

Flag	Bit
Topology Change (TC)	0
Proposal	1
Port Role	2
	3
Learning	4
Forwarding	5
Agreement	6
Topology Change ACK	7

Port Role:
00: Unknown
01: Alternate/Backup
10: Root
11: Designated

MST 802.1s

Features

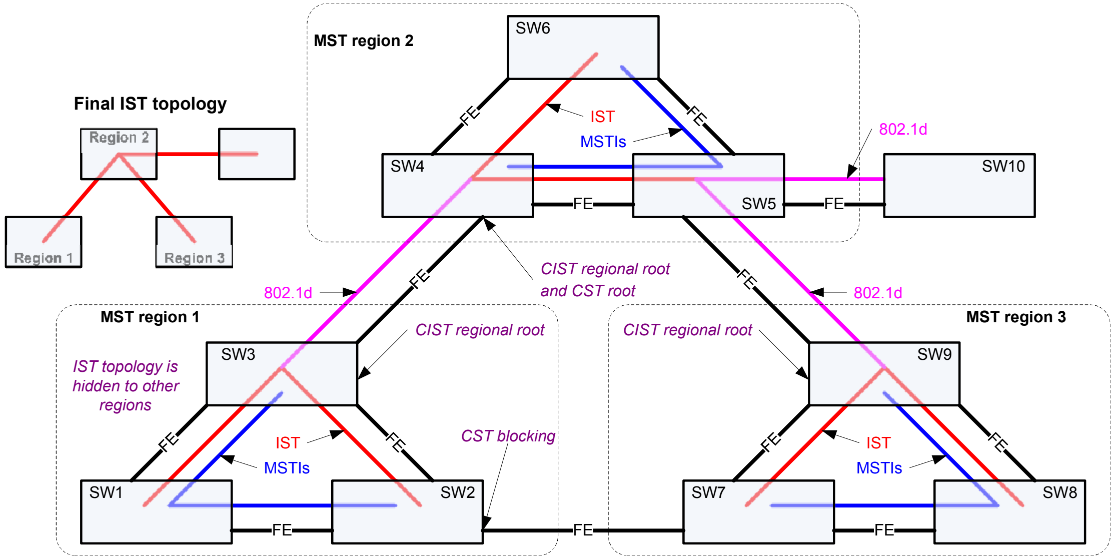
- Up to 16 MST (64 RFC) instances (no platform-specific limit for number of VLANs – max 4096) – there is always one instance 0 (undefined VLANs stay in it) + 15 user-defined. Instances can be numbered from 1 to 4096
- 802.1s introduces Regions (like AS in BGP) – switches in one common management. Switches belong to the same region if name, revision and vlans mappings are the same
- It is not recommended to have multiple regions. Place as many switches as you can inside one MST region. Migrate core (start with current root) and follow to access
- VLAN-to-instance mapping is not propagated. Only digest with region name and revision number is sent
- VLANs mapped to single MSTI must have the same topology (allowed VLANs on trunks). Avoid mapping VLANs to IST(0), and never manually prune individual VLANs (belonging to the same MSTI) from trunk
- 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
- 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. MST increases hop count of cascaded switches from 7 to 40 (20 is default) . It also uses 802.1t long cost mode to differentiate between GE, GEC, 10G.

Instances

- **IST (MSTI 0)** Internal Spanning Tree
  - The only instance that sends and receives BPDUs (even if no VLANs are assigned to MST0). 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. First implementation of pre-standard MISTP (Cisco proprietary MST) tunneled extra BPDUs across MST
  - It is recommended to have IST root inside MST. Successful MST and PVST+ interaction is possible if MST bridge is the root for all VLANs. If MST is the root for CTS and other switch (PVST+) is the root for any of the VLANs, boundary port will become root-inconsistent
  - Represents MST region as CST virtual bridge to outside. 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)
- **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
  - If the root bridge for CIST is within a non−MST region, the priority of VLANs 2 and above within that area must be better (smaller) than that of VLAN 1
  - If the root bridge for CIST is within a MST region, VLANs 2 and above in the non−MST area must have priorities worse (greater) than that in CIST root
- **MSTI** – Multiple Spanning Tree Instances (one or more) - RSTP instances within a region. RSTP is enabled automatically by default

Config

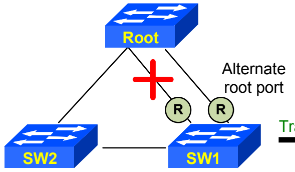
- ***spanning-tree mst configuration***
  name <name>
  revision <number>
  instance <id> vlan <range>
  Must be defined on every switch in region
- ***(G) spanning-tree mode mst***
  Configure on all switches AFTER all switches have consistent region configuration
- ***(G) spanning-tree mst <instance-id> root {primary | secondary}***
- ***(G) spanning-tree mst <instance-id> max-hops <count>***
- ***(G) spanning-tree mst <instance-id> <other STP parameters, timers>***
- ***(IF) spanning-tree mst pre-standard***
  If 802.1s and pre-standard MISTP ports are connected
- ***show spanning-tree mst ...***
- You can use VTPv3 to distibute VLAN-to-Instance mapping
  - ***(G) vtp primary mst***
    (G) vtp mode server mst
  - ***(G) vtp mode client mst***
  - You cannot configure MST manually if VTPv3 is running for MST propagation



show spanning-tree inconsistentports

STP

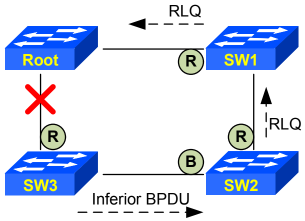
Uplinkfast



- 802.1d legacy feature 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 dummy frames 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
- Cannot be enabled on a switch that has STP priority modified
- ***(G) spanning-tree uplinkfast [max-update-rate <rate>]***
  If ***rate*** is 0 then no multicast flooding takes place (150 default)

Work only in legacy STP. Deactivated when RSTP is enabled

Backbonefast



- 802.1d legacy feature used for indirect link failure detection – explicit verification of inferior BPDUs. Recovery within 30 sec.
- ***(G) spanning-tree backbonefast***
  All switches within a domain must be configured
- If inferior BPDU is received on block port, switch SW2 sends proprietary Root Link Query messages on root and alternate (blocked upstream) ports containing SW2's root information and SW2 BID
- If upstream switch has the same root information as SW2 it forwards it to root ports. Root switch confirms it's still a root with positive answer flooding on all DP
- If any switch has different information, immediate negative answer is sent, and SW2 performs root election without waiting MaxAge (only Lisening and Learning). In case of positive answer blocked port changes to Listening and Learning

Portfast

- Immediately switches over to forwarding state. Avoid TCN generation for end hosts
- BPDU guard should be enabled on that port. Portfast does not turn off STP on that port
- ***(IF) spanning-tree portfast [trunk]***
  Trunk must be set if port is a trunk, otherwise, portfast does not work
- ***(G) spanning-tree portfast default***
  Enable portfast on all access ports (but not router trunks)
- ***(IF) switchport mode host***

UDLD

- Sends local port ID and remote (seen) port ID. Remote end compares with own state
- Unlike loopguard, UDLD protects against wrong wiring, and is per-physical-port, not per-VLAN
- Not really required on UTP ports, as Fast Link Pulses verify connectivity
- ***(G) udld message time <sec>***
  Default L2 probes sent every 15 sec to mac 01:00:0C:CC:CC:CC. Must be ACKed by remote end. Dead is 3x hello.
- Timers should be set, so link failure is detected before STP forward delay timer expires
- Normal mode does nothing except syslog (on some platforms it may err-disable port on the side where misconfiguration detected), and port is set to Undetermined state
- Aggresive mode attempts to reconnect once a second 8 times before err-disabling both ends
- If configured for the first time it is not enabled untill first Hello is heard from the other side
- ***(G) udld {enable | aggressive}***
  Enable UDLD in normal (***enable***) or aggresive mode only on all fiber-optic interfaces
- ***(IF) udld port [aggressive]***
  Enable UDLD in normal or aggressive mode on fiber-optic (override global mode) and twisted-pair link
- ***udld reset*** – reset err-disable state without shutting down port
- ***show udld [{<if> | neighbors}]***

BPDU guard

- Err-disable portfast port upon receiving BPDU
- ***(G) spanning-tree portfast bpduguard default***
  Applied only to interfaces which are in portfast state
- ***(IF) spanning-tree bpduguard enable***
- ***(G) errdisable detect cause bpduguard shutdown vlan***
  Prevent the port from shutting down, and shut down just the offending VLAN on the port where the violation occurred
- ***show interfaces status err-disabled***

BPDU filter

- ***(IF) spanning-tree bpdufilter enable***
  Port does not send any BPDUs and drops all BPDUs received (completely disables STP). Applies to any interface. Do not use! Can cause loops. Takes precedence over bpduguard, so bpduguard has no chance to err-disable the port
- ***(G) spanning-tree portfast bpdufilter default***
  Applies only to interfaces in portfast state. Sends 11 BPDUs on port activation or upon receiving BPDU. Does not filter deceived BPDUs. Portfast state changes to non-portfast upon receiving BPDU. Does not cause loops
- The interfaces still send a few BPDUs at link-up before the switch begins to filter outbound BPDUs

Etherchannel guard

- ***(G) spanning-tree etherchannel guard misconfig***
  Enabled by default. Uses BPDU, if it comes back on a port, meaning one of etherchannel ports on remote end is not in common channel
- If etherchannel is not detected all bundling ports go into err-disable.
- A misconfiguration can occur if local interfaces are configured in an EtherChannel, but the interfaces on the other device are neither LACP, PAgP, nor ON.

Root guard

- Can be enabled on **designated ports only**. Opposite to loop guard
- When superior BPDU is received on a DP, the port becomes root-inconsistent. Recovery after ForwardDelay sec of not receiving superior BPDU
- 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***
- ***show spanning-tree inconsistentports***

Loop guard

- If no BPDUs are received on a blocked port for a specific length of time (MaxAge 20 sec), Loop Guard puts that port (per VLAN) into loop-inconsistent blocking state, rather than transitioning to forwarding state
- Unlike UDLD, loopguard protects against STP software problems (bugs, etc)
- Can be enabled on **non-designated ports only**, which are root and alternate ports (no effect on other ports). Cannot be enabled on portfast and dynamic VLAN ports. Enabling on shared links is highly not recommended.
- Automatic recovery when BPDU is again received
- ***(G) spanning-tree loopguard default***
- ***(IF) spanning-tree guard loop***

Bridge Assurance

- Permanent, bi-directional BPDU exchange, regardless of both sides' port state, replacement for loopguard
- Runs in RSTP or MST only. Err-disables (*BA_Inc) port when it stops seeing BPDU
- Since it runs per VLAN, it prunes VLANs which are not configured on neighbor switch (no BPDU received)
- ***(G) spanning-tree bridge assurance***
  Enabled by default. Disabling BA causes all ports to behave as normal spanning tree ports
- ***(IF) spanning-tree portfast network***
  Enable/disable BA per port

Dispute

- Always enabled, cannot be disabled (no commands)
- Protects against software issues (bug) – BPDU with DP role received on the port which also has DP role

Port Channel

Features

In Layer 2 EtherChannels, the first port in the channel that comes up provides its MAC address to the EtherChannel. If this port is removed from the bundle, one of the remaining ports in the bundle provides its MAC address to the EtherChannel.

For Layer 3 EtherChannels, the MAC address is allocated by the stack master as soon as the interface is created

Speed for one flow is still limited to the speed of one link (load-balancing), unlike MLPPP

All physical interfaces must have identical configuration. If any of speed, duplex, trunking mode, allowed vlans is different, the port is not bound to etherchannel. STP costs does not have to be the same on physical interfaces

LACP or PAgP check links consistency. If They are disabled, inconsistency (STP loop) can occur (Etherchannel on one side, single links on other side)

(Po1) no switchport – create L3 port-channel

(Po1) port-channel min-links <#>
By default, etherchannel is active as long as at least one link is active. STP cost is not adjusted when links go down. You can make sure that data flow chooses hi-bandwidth redundant path in case only few links are left.

(IF) channel-group <id> mode on
Manual port-channel does not respond to neither PAGP, nor LACP

(G) port-channel load-balance {dst-ip | dst-mac | src-dst-ip | src-dst-mac | src-ip | src-mac}
Set the load-distribution method among the ports. Src-mac is default (XOR on rightmost bits of MAC)

Always use „power of 2" number of links for port-channels

Links	Hash
8	1:1:1:1:1:1:1:1
7	2:1:1:1:1:1:1
6	2:2:1:1:1:1
5	2:2:2:1:1
4	2:2:2:2
3	3:3:2
2	4:4

Verify

show etherchannel load-balance

show etherchannel {summary | detail | port-channel | protocol}

show interface etherchannel

VSS

Multi-chassis Etherchannel technology available on Cat 6500 (Virtual Switching System). Requires min. Sup-720

Access switch is not aware of two chassis. Port-channel configuration is classical

One control plane (single configuration). NSF/SSO (RPR) – one chassis is active control, second is standby

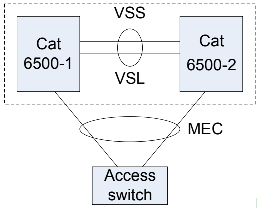
Two data planes (both switches pass traffic from L2 only etherchannel members, no STP blocking ports)

New interface naming: <chassis>/<module>/<if>

No need to use FHRP (HSRP, VRRP, GLBP)

Active chassis runs STP. Standby redirects BPDUs across the VSL to the active chassis

Init: 1) read config 2) start VSL 3) start VSLP 4) start redundancy RRP/SSO 5) boot system



VSL

Virtual Switch Link – port-channel (preferred) used for state sync and traffic flow

Requires 10G links (preferred port-channel)

Split-brain is avoided with: 1) Enhanced PaGP through access switches 2) separate L3 BFD link 3) separate L2 Fast Hello Dual Active Detection link

Frames forwarded over the VSL are encapsulated with a special 32-byte header

If possible, ingress traffic is forwarded to an outgoing interface on the same chassis, to minimize traffic on VSL

(Po X) switch virtual link 1
Identify VSL on switch 1

(Po Y) switch virtual link 2
Identify VSL on switch 2

(#) switch convert mode virtual
Perform on both switches

Virtual Switch Link Protocol (VSLP)

Role Resolution Protocol - negotiate the role (VSS active or VSS standby) for each chassis

Link Management Protocol - exchanges information required to establish communication

(G) switch virtual domain <id>
Domain must be the sam on both switches

(VSS) switch {1 | 2} priority <#>
Priority 1-255 (default 100), higher better – assumes active role

Switchover

(#) redundancy reload peer

(#) redundancy force-switchover

Verify

show switch virtual [{role | link}]

Cisco PAgP

Up to eight interfaces

In auto-negotiation mode it may take 15 sec to form EC. It takes place before STP. Negotiation should be disabled for hosts (off)

(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

Auto mode initiates session, desirable is silent and waits for initiation

(G) pagp learn-method {aggregation-port | physical-port}
How to learn the source address of incoming packets received from (aggr-port is default). If phy-port is used, then frames are sent always on the same port where MAC was learned.

(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

PAGP	LACP	Behavior
on	on	No dynamic negotiation. Forced.
off	off	PortChannel **negotiation** disabled
auto	passive	Wait for other side to initiate
desirable	active	Initiate negotiation

IEEE 802.3ad LACP

LACP protocol can run only on full-duplex ports

16 ports can be selected, but only max 8 is used. Rest is in hot-standby

Switch with lowest system priority makes decisions about which ports participate in bundling

(IF) channel-protocol lacp

(IF) channel-group <1-64> mode {passive | active}

(IF) lacp port-priority <#>
Priority decides which ports are used for EC, and which remain in standby.
Default 32768, lower is better. If priority is the same, Port ID is used (lower better)

(G) lacp system-priority <#>
The system priority (lower better) is used in conjunction with the MAC to form the system identifier

show lacp sys-id

show lacp neighbor

StackWise

Available on access platforms. Members must be the same platform

One control plane is synchronized over dedicated Stack cable (loop) on the back

Stack can have more than one member (9 on 3750X)

The switch with the highest priority becomes the new stack master when current master goes down (non-preemptive). If priority is the same then switch with no default interface-level configuration, highest IOS feature set, lowest MAC

The bridge ID and router MAC address are determined by the MAC address of the stack master.

(G) stack-mac persistent timer <min>
When the persistent MAC is enabled, the stack MAC address changes in specified time (default 4 min.) when master is down. If the previous master rejoins, the stack continues to use its MAC, even if the switch is now a plain member. If 0 is used, MAC never changes

Each stack member has a copy of running config

Never add powered-on switch to the stack, as new master can be elected and renumbering occurs (all switches reload) and new master's config is used. Power off first (when adding or removing)

Stack members that are powered on within 120-sec participate in the stack master election (can become the stack master). Members powered later do not participate in the election and become stack members

(G) switch <#> renumber <#>

(G) switch <#> priority <1-15> - default is 1

(#) reload slot <#> - required after priority is changed

(G) switch <#> provision <model> - preprovision offline switch

(#) session <#> - connect directly to the member

(#) remote command {all | <#>}

(#) switch <#> stack port <port-#> {disable | enable}
Use when stack is flapping. Stack will operate in half speed

show switch stack-ports summary

show switch

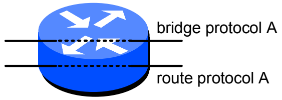
Bridging

Transparent Bridging

- Complies with the IEEE 802.1D standard
- ***(G) bridge <bridge-group> protocol ieee***
- ***(IF) bridge-group <bridge-group>***
- ***(G) bridge <bridge-group> acquire***
  Forward frames according to dynamicaly learned MAC addresses. If disabled, static mappings must be used
- ***(G) bridge <bridge-group> address <mac-address> {forward | discard} [<intf>]***
  Filter frames with a specific source or destination MAC address
- ***interface bvi <bridge-group>***
  ip address ...
  Create L3 interface representing the bridge group on the router

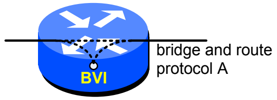
Concurrent Routing and Bridging

- Route 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
- ***(G) bridge crb***
- ***(G) bridge <bridge-group> route <protocol>***
  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



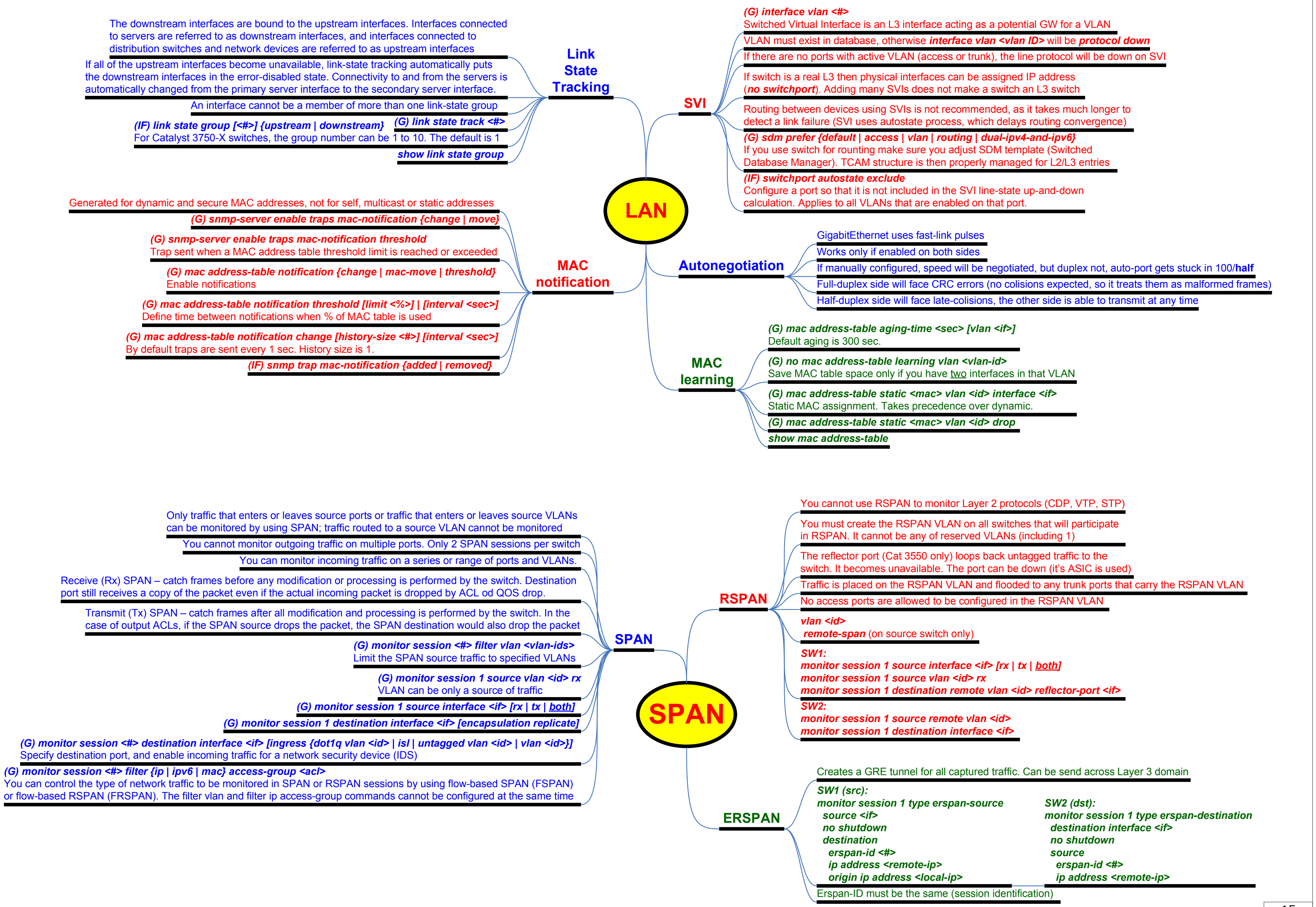
Integrated Routing and Bridging

- Routers do not support per-vlan STP, so Bridge Priority is always 32768 for every VLAN, which is lower than any value on switches, which add VLAN id, so router will be a root for all VLANs by default
- 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 (BVI) 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.
- ***(G) bridge irb***
- ***(G) interface bvi <bridge-group>***
- ***(G) bridge <bridge-group> route <protocol>***
- ***(G) bridge <bridge-group> bridge <protocol>***



Fallback Bridging

- With fallback bridging, the switch bridges together two or more VLANs or routed ports, connecting multiple VLANs within one bridge domain. Useful when you have two separate VLANs and subnets but need to bridge non-routable protocol between the two VLANs
- Fallback bridging does not allow spanning trees from VLANs to collapse. Each VLAN has own SPT instance. There is also separate SPT, called VLAN-bridge SPT, which runs on top of the bridge group to prevent loops
- ***(IF) bridge-group <#> spanning-disabled***
  Disable spanning tree on the port. BPDUs can be prevented from traveling through the router across the WAN link.
- ***(G) bridge <#> protocol vlan-bridge***
- ***(IF) bridge-group <#>***
  Assign bridge to interface VLAN
- ***(IF) bridge-group <#> priority <#>***
  Port priority for interface VLAN
- ***(IF) bridge-group <#> path-cost <#>***
  Path cost for interface VLAN
- ***1) no bridge <group> acquire***
  2) bridge <group> address <mac> {forward | discard} [<interface>]
  By default, switch forwards any frames it has dynamically learned. The switch can forward only frames whose MAC addresses are statically configured (static MAC for bridge, not for mac-address-table !!!).



LAN

Link State Tracking

- The downstream interfaces are bound to the upstream interfaces. Interfaces connected to servers are referred to as downstream interfaces, and interfaces connected to distribution switches and network devices are referred to as upstream interfaces
- If all of the upstream interfaces become unavailable, link-state tracking automatically puts the downstream interfaces in the error-disabled state. Connectivity to and from the servers is automatically changed from the primary server interface to the secondary server interface.
- An interface cannot be a member of more than one link-state group
- ***(G) link state track <#>***
- ***(IF) link state group [<#>] {upstream | downstream}***
  For Catalyst 3750-X switches, the group number can be 1 to 10. The default is 1
- ***show link state group***

MAC notification

- Generated for dynamic and secure MAC addresses, not for self, multicast or static addresses
- ***(G) snmp-server enable traps mac-notification {change | move}***
- ***(G) snmp-server enable traps mac-notification threshold***
  Trap sent when a MAC address table threshold limit is reached or exceeded
- ***(G) mac address-table notification {change | mac-move | threshold}***
  Enable notifications
- ***(G) mac address-table notification threshold [limit <%>] | [interval <sec>]***
  Define time between notifications when % of MAC table is used
- ***(G) mac address-table notification change [history-size <#>] [interval <sec>]***
  By default traps are sent every 1 sec. History size is 1.
- ***(IF) snmp trap mac-notification {added | removed}***

SVI

- ***(G) interface vlan <#>***
  Switched Virtual Interface is an L3 interface acting as a potential GW for a VLAN
- VLAN must exist in database, otherwise ***interface vlan <vlan ID>*** will be ***protocol down***
- If there are no ports with active VLAN (access or trunk), the line protocol will be down on SVI
- If switch is a real L3 then physical interfaces can be assigned IP address (***no switchport***). Adding many SVIs does not make a switch an L3 switch
- Routing between devices using SVIs is not recommended, as it takes much longer to detect a link failure (SVI uses autostate process, which delays routing convergence)
- ***(G) sdm prefer {default | access | vlan | routing | dual-ipv4-and-ipv6}***
  If you use switch for rounting make sure you adjust SDM template (Switched Database Manager). TCAM structure is then properly managed for L2/L3 entries
- ***(IF) switchport autostate exclude***
  Configure a port so that it is not included in the SVI line-state up-and-down calculation. Applies to all VLANs that are enabled on that port.

Autonegotiation

- GigabitEthernet uses fast-link pulses
- Works only if enabled on both sides
- If manually configured, speed will be negotiated, but duplex not, auto-port gets stuck in 100/**half**
- Full-duplex side will face CRC errors (no colisions expected, so it treats them as malformed frames)
- Half-duplex side will face late-colisions, the other side is able to transmit at any time

MAC learning

- ***(G) mac address-table aging-time <sec> [vlan <if>]***
  Default aging is 300 sec.
- ***(G) no mac address-table learning vlan <vlan-id>***
  Save MAC table space only if you have two interfaces in that VLAN
- ***(G) mac address-table static <mac> vlan <id> interface <if>***
  Static MAC assignment. Takes precedence over dynamic.
- ***(G) mac address-table static <mac> vlan <id> drop***
- ***show mac address-table***

SPAN

SPAN

- Only traffic that enters or leaves source ports or traffic that enters or leaves source VLANs can be monitored by using SPAN; traffic routed to a source VLAN cannot be monitored
- 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
- ***(G) monitor session <#> filter vlan <vlan-ids>***
  Limit the SPAN source traffic to specified VLANs
- ***(G) monitor session 1 source vlan <id> rx***
  VLAN can be only a source of traffic
- ***(G) monitor session 1 source interface <if> [rx | tx | both]***
- ***(G) monitor session 1 destination interface <if> [encapsulation replicate]***
- ***(G) monitor session <#> destination interface <if> [ingress {dot1q vlan <id> | isl | untagged vlan <id> | vlan <id>}]***
  Specify destination port, and enable incoming traffic for a network security device (IDS)
- ***(G) monitor session <#> filter {ip | ipv6 | mac} access-group <acl>***
  You can control the type of network traffic to be monitored in SPAN or RSPAN sessions by using flow-based SPAN (FSPAN) or flow-based RSPAN (FRSPAN). The filter vlan and filter ip access-group commands cannot be configured at the same time

RSPAN

- You cannot use RSPAN to monitor Layer 2 protocols (CDP, VTP, STP)
- You must create the RSPAN VLAN on all switches that will participate in RSPAN. It cannot be any of reserved VLANs (including 1)
- The reflector port (Cat 3550 only) loops back untagged traffic to the switch. It becomes unavailable. The port can be down (it's ASIC is used)
- Traffic is placed on the RSPAN VLAN and flooded to any trunk ports that carry the RSPAN VLAN
- No access ports are allowed to be configured in the RSPAN VLAN
- ***vlan <id>***
  remote-span (on source switch only)
- ***SW1:***
  monitor session 1 source interface <if> [rx | tx | both]
  monitor session 1 source vlan <id> rx
  monitor session 1 destination remote vlan <id> reflector-port <if>
- ***SW2:***
  monitor session 1 source remote vlan <id>
  monitor session 1 destination interface <if>

ERSPAN

- Creates a GRE tunnel for all captured traffic. Can be send across Layer 3 domain

```
SW1 (src):
monitor session 1 type erspan-source
 source <if>
 no shutdown
 destination
  erspan-id <#>
  ip address <remote-ip>
  origin ip address <local-ip>
```

```
SW2 (dst):
monitor session 1 type erspan-destination
 destination interface <if>
 no shutdown
 source
  erspan-id <#>
  ip address <remote-ip>
```

- Erspan-ID must be the same (session identification)

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

```
R1#sh ip cef
Prefix               Next Hop             Interface
0.0.0.0/0            no route
2.2.2.2/32           10.0.12.2            GigabitEthernet0/0
10.0.12.0/24         attached             GigabitEthernet0/0
10.0.12.0/32         receive              GigabitEthernet0/0
10.0.12.1/32         receive              GigabitEthernet0/0
10.0.12.2/32         attached             GigabitEthernet0/0
10.0.12.255/32       receive              GigabitEthernet0/0

R1#sh ip cef 2.2.2.2 detail
2.2.2.2/32, epoch 0
  1 RR source [no flags]
  recursive via 10.0.12.2
    attached to GigabitEthernet0/0
```

Static route NH

Static route

CEF

Features

- Route Caching – demand base lookup. CEF – topology based lookup
- IOS will switch a packet using CEF only if CEF is enabled on the inbound interface (not outbound)
- Cache building is not triggered by the first packet, but for all entries in a routing table. All changes in routing table are automaticaly reflected in FIB
- RIB – Routing Information Base. Routing table populated by routing protocols
- FIB – Forwarding Information Base. Populated by RIB. Topology-driven 8-8-8-8 mtrie
- Adjacency Table – L2 table of adjacent neighbors (next-hop)
- *(G) ip cef [distributed]*
- *(IF) ip route-cache cef*

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 intf is not directly known via routing protocol from which prefix was received. Recursive lookup required
- ***show ip cef [vrf <name>] [<ip>] [detail] [internal]***
  CEF is built independently for global routing and each VRF

Adjacency Table

- Contains all connected next-hops, interfaces and associated L2 headers
- glean – Destination is attached via broadcast network but MAC is yet unknown. Individual host adjacency in addition to whole prefix entry
- 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
- null – Pointed to Null0
- ***show adjacency [detail]***
  Routes associated with outgoing interface and L2 header

```
R1#show adjacency detail
Protocol Interface                 Address
IP       GigabitEthernet0/0        10.0.12.2(13)
                                   0 packets, 0 bytes
                                   epoch 0
                                   sourced in sev-epoch 0
                                   Encap length 14
                                   CA020FF00008CA0108CC00080800
                                   L2 destination address byte offset 0
                                   L2 destination address byte length 6
                                   Link-type after encap: ip
                                   ARP
```

Number of times that this adjacency is pointed to by FIB entries

All entries for which L2-L3 mappings are known

Ethernet

L2-L3 mapping protocol

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 #)*
- ***show ip cef exact-route <src> <dst>***
  Check which path IPv4 packet will take
- Polarization
  - Hash algorithm chooses particular path and the redundant paths remain completely unused
  - To avoid polarization different hashing algorithms can be used on different layers (core, dist)
  - Universal algorithm, using universal-ID (randomly generated at the boot up), adds a 32-bit router-specific value to the hash function. Ensures that the same src/dsi pair hash into a different value on different routers
  - ***(G) ip cef load-sharing algorithm universal <id>***
  - Does not work for an even number of equal-cost paths due to a hardware limitation. IOS adds one artificial link to adjacency table when there is an even number of equal-cost paths to make calculations more efficient

IOS-XE

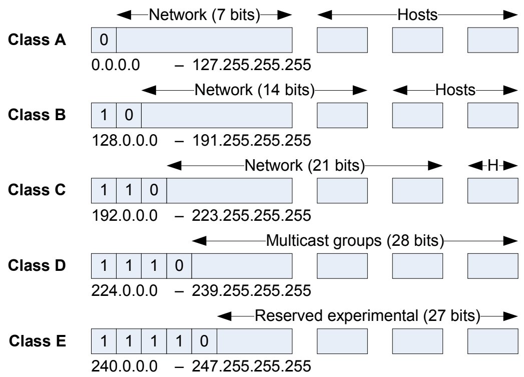
Packages

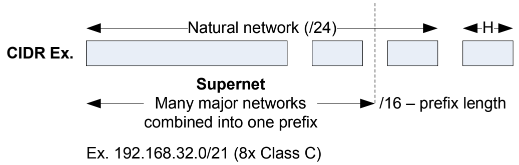
- Consolidated packages and optional subpackages. Can be updated as a whole OS or individually
- RPBase – Base functionality(OS) of route processor
- RPControl – Control-plane processes that interface between IOS and the rest of the platform
- RPAccess – Remote access (SSH, SSL)
- RPIOS – Routing and forwarding (15.x IOS) on RP
- ESPBase – Embedded Services Processor operating system, control processes
- SIPBase – SPA Interface Processor operating system, and control processes
- SIPSPA – Shared Port Adapters drivers and field-programmable device (FPD)

Managers

- Forwarding and Feture Manager
  - Separation of Control Plane and Data Plane
  - Programs Data Plane with Forwarding Engine Driver
- Forwarding Manager
- Forwarding Engine Driver – Provided by the platform instantiation of hardware driver
- Chassis Manager – HA functions
- Host Manager
- Interface Manager
- Shell Manager
- Logger

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





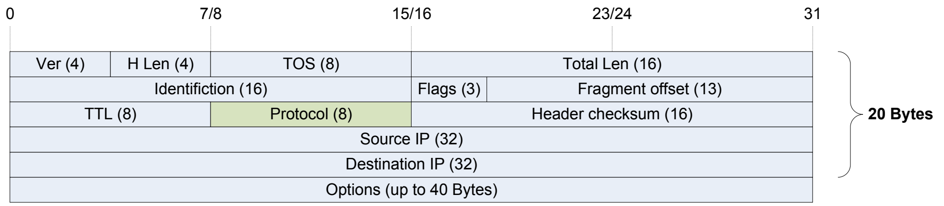
Ex. 192.168.32.0/21 (8x Class C)

Common networks

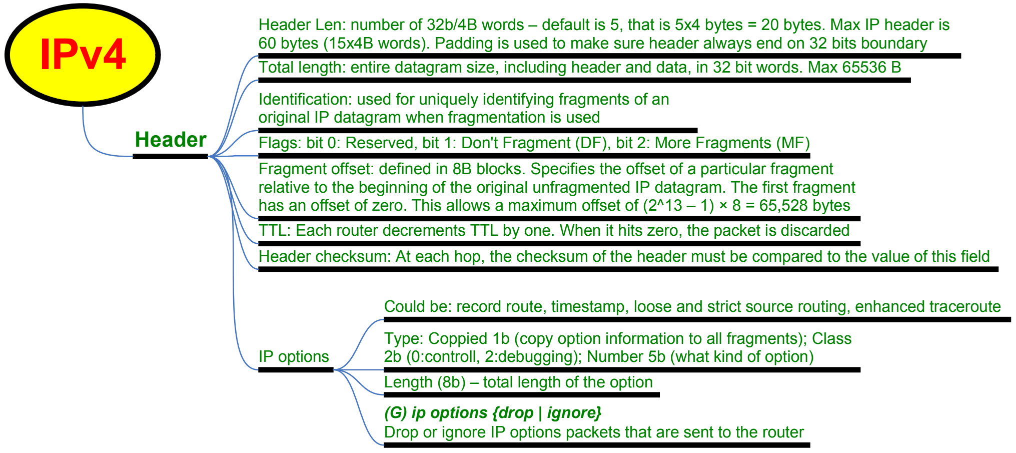
0.0.0.0/8	Default network
10.0.0.0/8	Private network
127.0.0.0/8	Loopback
169.254.0.0/16	Link-Local
172.16.0.0/12	Private network
192.0.0.0/24	Reserved (IANA)
192.0.2.0/24	Test network
192.88.99.0/24	IPv6 to IPv4 relay
192.168.0.0/16	Private network
198.18.0.0/15	Network benchmark tests
198.51.100.0/24	Test network
203.0.113.0/24	Test network
224.0.0.0/4	Multicasts
240.0.0.0/4	Reserved
255.255.255.255	Broadcast

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
102	HSRPv2
103	PIM
112	VRRP



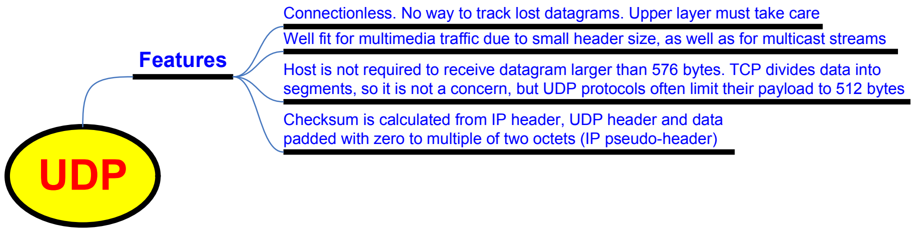
Ver (4)	H Len (4)	TOS (8)	Total Len (16)	
Identification (16)			Flags (3)	Fragment offset (13)
TTL (8)		Protocol (8)	Header checksum (16)	
Source IP (32)				
Destination IP (32)				
Options (up to 40 Bytes)				



IPv4

Header

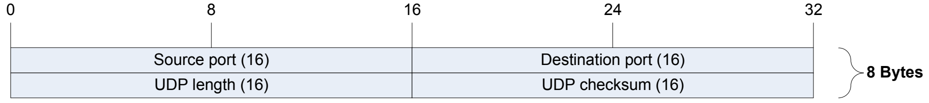
- Header Len: number of 32b/4B words – default is 5, that is 5x4 bytes = 20 bytes. Max IP header is 60 bytes (15x4B words). Padding is used to make sure header always end on 32 bits boundary
- Total length: entire datagram size, including header and data, in 32 bit words. Max 65536 B
- Identification: used for uniquely identifying fragments of an original IP datagram when fragmentation is used
- Flags: bit 0: Reserved, bit 1: Don't Fragment (DF), bit 2: More Fragments (MF)
- Fragment offset: defined in 8B blocks. Specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram. The first fragment has an offset of zero. This allows a maximum offset of (2^13 – 1) × 8 = 65,528 bytes
- TTL: Each router decrements TTL by one. When it hits zero, the packet is discarded
- Header checksum: At each hop, the checksum of the header must be compared to the value of this field
- IP options
  - Could be: record route, timestamp, loose and strict source routing, enhanced traceroute
  - Type: Coppied 1b (copy option information to all fragments); Class 2b (0:controll, 2:debugging); Number 5b (what kind of option)
  - Length (8b) – total length of the option
  - ***(G) ip options {drop | ignore}***
    Drop or ignore IP options packets that are sent to the router



UDP

Features

- Connectionless. No way to track lost datagrams. Upper layer must take care
- Well fit for multimedia traffic due to small header size, as well as for multicast streams
- Host is not required to receive datagram larger than 576 bytes. TCP divides data into segments, so it is not a concern, but UDP protocols often limit their payload to 512 bytes
- Checksum is calculated from IP header, UDP header and data padded with zero to multiple of two octets (IP pseudo-header)



Source port (16)	Destination port (16)
UDP length (16)	UDP checksum (16)

TCP

Header

Offset: TCP header length. The same rules apply as for IP header

Initial SNs for new sessions start with 1 and increments every 0.5 sec and at every new connection by 64000, cycling to 0 after about 9,5h. The reason for this is that each connection starts with different initial numer

Flags
1 bit each

- CWR – Congestion Window Reduced flag is set by the sending host to indicate that it received a TCP segment with the ECE flag set and had responded in congestion control mechanism
- ECE – Explicit Congestion Notification (ECN-Echo) – not the same as ECN in IP header TOS field
- URG – indicates that the Urgent pointer field is significant
- ACK – Acknowledges data received. All packets after the initial SYN should have this flag set
- PSH – Asks to immediately push the buffered data to the receiving application. Normally, TCP waits for the buffer to exceed the MSS – can be probematic (delay) for applications sending small data
- RST – Reset the connection
- SYN – Exchange sequence numbers. Only the first packet sent from each end should have this flag set
- FIN – No more data from sender, connection can be closed

(G) ip tcp window-size <bytes>
Window size: defines the number of bytes receiver is willing to accept before it sends ACK. Initially set to number of bytes set as ACK SN sent in 3-way handshake. Default is 4128 B

Options can be MSS, Timestamp, Selective ACK. It is exchanged only in first segments (SYN)

(G) ip tcp selective-ack
TCP might not experience optimal performance if multiple packets are lost from one window of data. Receiver returns selective ACK packets to sender, informing about data that has been received. The sender can then resend only the missing data segments

(G) ip tcp timestamp
TCP time stamp improves round-trip time estimates

Connection

3-way handshake is required before data can be sent. Each side sets own SN independently, and exchanges it with the other side

Closing connection is a 4-way. Any endpoint can send FIN to signal EoT, it must be ACKed. Since TCP is a full-duplex, other side must also send FIN and wait for ACK

(G) service tcp-keepalive {in | out}
Detect dead sessions (probe idle connections)

(G) ip tcp synwait-time <sec>
Timeout for establishing all TCP sessions from a router. Default is 30 sec. Can be used to speed up telnet timeout for non-responding hosts

show tcp brief all [numeric]

show tcp tcb <#>
Show detailt TCP session information. Acquire TCP from ***show tcp brief all***

MSS

(G) ip tcp mss <#>
Define MSS for TCP connections from and to a router. Default is 1460 for local destination (without IP and TCP headers), or 536 for remote

TCP is a stream protocol, unlike UDP, where each write, performed by application, generates separate UDP segment. TCP collects writes and may send them all in one segment as chunks

MSS is a largest amount of **data (without headers)** that TCP is willing to send in a single segment. **MSS = MTU – IP header – TCP header**. Should be small enough to avoid fragmentation

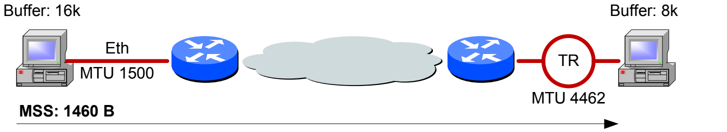
Derived from local interface MTU minus TCP and IP headers. (Ex. 1460 for ethertnet). Sender compares own MSS and local MTU, chooses lower one and sends this MSS to receiver

When destination IP is non-local or other side does not set MSS, then MSS is set to 536 (20B IP and 20B TCP is added, so IP packet fits into min 576B required by RFC for host to accept)

Received MSS is always compared only to local MTU – smaller value is used. If there is smaller MTU somewhere on the path, fragmentation will occur. PMTUD should be used to find lowest MTU on the path (tuneling on intermediate routers lowers MTU)

0 7/8 15/16 23/24 31

Source port (16)		Destination port (16)		20 Bytes
Sequence number (32)				
Acknowledgement sequence number (32)				
Offset (4)	Reserved (4)	Flags (8)	Window Size (16)	
TCP Checksum (16)		Urgent pointer (16)		
Options (max 40 bytes)				



MSS: 1460 B

1) Host A compares buffer 16K to local MTU 1460 (MTU – headers), and uses lower value: 1460 B

2) Host B compares received MSS 1460 with local MTU (excluding headers, for sake of MSS calculation) 4422. Uses lower value: 1460 B

3) Payload: 1460 B

MSS: 4422 B

4) Host B compares buffer 8K to local MTU 4422 (MTU – headers), and uses lower value: 4422 B

5) Host A compares received MSS 4422 with local MTU (excluding headers, for sake of MSS calculation) 1460. Uses lower value: 1460 B

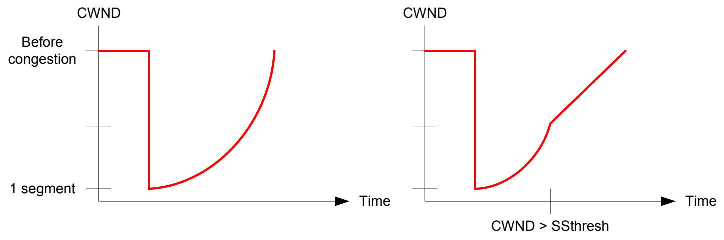
6) Payload: 1460 B

Common port numbers					
echo	7/tcp/udp	nntp	119/tcp	dhcpv6 (client)	546/tcp/udp
discard	9/tcp/udp	ntp	123/udp	dhcpv6 (server)	547/tcp/udp
daytime	13/tcp/udp	netbios-ns	137/tcp/udp	ldp	646/udp
chargen	19/tcp/udp	netbios-dgm	138/tcp/udp	iscsi	860/tcp
ftp-data	20/tcp	netbios-ssn	139/tcp/udp	imap-ssl	993/tcp
ftp	21/tcp	imap	143/tcp	h323	1720/udp
ssh	22/tcp	snmp	161/udp	h323	1721/tcp
smtp	25/tcp	snmptrap	162/udp	radius-auth	1812/udp
tacacs	49/tcp	bgp	179/tcp	tadius-acct	1813/udp
dns	53/tcp/udp	ldap	389/tcp/udp	sccp	2000/udp
bootps (server)	67/udp	https	443/tcp	mdcp	2427/udp
bootpc (client)	68/udp	ms-ad	445/tcp	iscsi-targe	3260/tcp
tftp	69/udp	isakmp	500/udp	rdp	3389/tcp/udp
http	80/tcp	syslog	514/udp	ipsec-nat	4500/udp
pop3	110/tcp	rip	520/udp	sip	5060/tcp
auth	113/tcp/udp	ripng	521/udp	sip-tls	5061/tcp

TCP

Congestion

- TCP uses also congestion window (CWND). It is not communicated between peers. TCP sender calculates CWND by its own - varies in size much more quickly than advertised window as it reacts to congestion
- TCP sender always uses the lower of the two windows to determine how much data it can send before receiving ACK
- 3-way handshake sets CWND = 1 and Slow Start Threshold (SSTHRESH) = 65535
- **1)** TCP sender fails to receive ACK in time (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
- CWND grows at an exponential rate during slow start
- Congestion avoidance allows CWND to grow slower at a linear rate
- Congestion Avoidance and Slow Start are algorithms with different objectives. In practice they are implemented together



Slow-start only

Slow-start and congestion avoidance

Flow Control

- Receiver specifies the receive window with the amount of data it is willing to buffer. Sending host can send only up to that amount of data before it waits for an acknowledgment
- Windows is set in every segment, and is floating, depending on how fast process reads data from incoming buffer. ACK can set window to 0, which means receiver's process hasn't read data from buffer yet. A while later ACK is sent with updating window. It looks like another ACK but it's just Window Update
- Sender does not have to fill whole receiver's window. Receiver does not have to wait until whole window is filled
- Persist Timer is started after each window=0. Window Probe is sent after timer expires (no Window Update was received) at 60 sec intervals until session is terminated or new windows is advertised. To avoid sending small segments while buffer is being freed (silly window syndrome), receiver does not advertise new window until half of available buffer is free
- TCP usually does not send ACK at the same time data has been received. It waits (200ms) so maybe some data can be send back (piggyback ACK). If there is data to be sent ACK is send immediately. The 200ms timer goes off at fixed intervals, so ACK can wait from 1 to 200 msec, depending on when data was received
- ***(G) service nagle***
  Nagle – collect data and send them in one segment to avoid tinygrams (for poorly implemented applications – flows). It is not recommended for interactive applications (mouse movements).

BW-Delay Product

- Data link's capacity (in bps) and its round-trip delay time (in sec.) - maximum amount of data on the network circuit at any given time (data that has been transmitted but not yet acknowledged)
- LFN (Long Fat Network) – pipe with high bandwidth, but also large delay – large BW-DLY value
- The solutions for LFN is to have bigger TCP window

0 7/8 15/16 23/24 31

Type (8)	Code (8)	Checksum (16)
Rest of header data (8B) – depends on context		

12 Bytes

ICMP

Features

- Error message is never send if another error message is received to avoid loops. Also, it is never sent in reply to broadcast or multicast packets or other IP segments than first, as well as packets with source IP not defining single host (loopback, broadcast, all zeros, etc)
- Error messages must include in their payload original IP header with all options and first 8 bytes of data following IP header in original packet. IP header allows to interpret those 8 bytes od data. For TCP and UDP ports are included in those 8 bytes, so for example unreachable can be generated stating which ports are unreachable
- ICMP echo contains identifier which allows to distinguish between several processes sending ping message from single host. Also sequence number is included, starting from 0 incrementing by 1 with every message sent.
- Record Route IP option stores max 9 hops. 20 bytes fixed IP header, 40 bytes left, 3 used for IP option overhead – own header, then 37 bytes available. Each IP address is 4 bytes, so 9 hops = 36 B is used.

Unreachable

- For unreachable message first 32 bits in ICMP payload are unused (all 0), they can be used to define MTU for PMTUD mechanism.
- ***(IF) no ip unreachables***
  By default enabled. Affects all types of ICMP unreachable messages (traceroute, etc)
- Type 3
  - Code 0: Network unreachable (no route)
  - Code 1: Host unreachable (no host L2 address on end router)
  - Code 2: Protocol unreachable
  - Code 3: Port unreachable
  - Code 4: Fragmentation needed, but DF set

Redirect

- Redirect contains in reserved 4 octets gateway IP address of router to be used for sending packets to a destination network. Redirects can be generated only by routers, not hosts. Also, routers do not use redirect messages, they use routing table
- ***(IF) no ip redirects***
  By default enabled. Enable sending of ICMP redirect messages if routing for destination points through the same interface on which packet was received
- Type 5
  - Code 0: Network redirect
  - Code 1: Host redirect

Traceroute

- It sends UDP messages with dest port most likely not being used (above 30000). Intermediate hosts send Time Exceeded, but when datagram reached end host, even if TTL is 1, it does not generate Time Exceedd (as it is a final host), so Port Unreachable is generated
- Hosts which receive datagram with TTL 0 or 1 must NOT forward it. If TTl=0 they drop it and sent Time Exceeded ICMP message

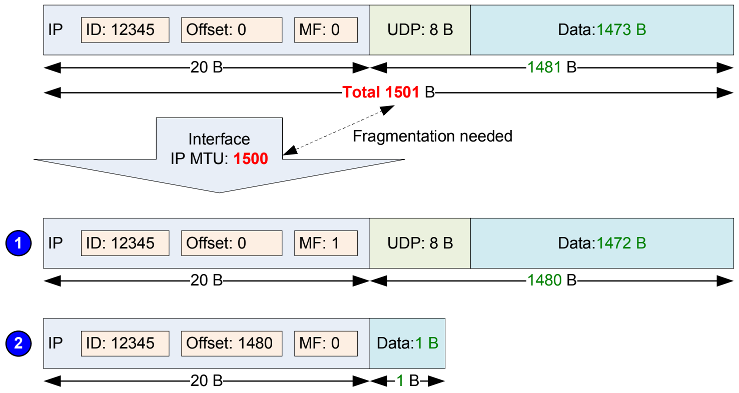
Ping

- Output: „!" - OK. „." - (dot) timeout, „M" - usualy fragmentation needed but DF set, „U" - unreachable

MTU

Fragment

- Maximum datagram length is 65k, but most links enforce lower MTU. IP packets can be fragmented to alleviate MTU differences.
- When IP datagram is fragmented, it is not reassembled until it reaches final host (or router in case of tunnel endpoint if tunneled traffic is fragmented)
- Dropped fragments cause whole IP packet t be retransmitted
- Components in IP header
  - 16 bit identifier identifies whole datagram. It is the same in all fragments.
  - DF - used by PMTUD, 0:may fragment, 1:don't fragment
  - MF - 0:last fragment, 1:more fragments
  - 13 bits fragment offset (in Bytes). First fragment starts with 0
- IP header (20 bytes) is added to each fragment. Original IP datagram size can be determined only after last fragment is received
- Fragmentation is problematic for receiver. Hosts don't have problems, as they have resources for this. Router reserves maximum available buffer for fragmented packet, as it has no idea how large the packet will be. This consumes scarce resources



PMTUD

- *(G) ip tcp path-mtu-discovery [age-timer {<min> | infinite}]*
  Enable PMTUD. Default time is 10 min. It changes the default MSS to 1460 even for nonlocal nodes.
- PMTUD is supported only for TCP traffic and is independent in both directions
- If host supports PMTUD (in most cases it does), all packets have DF bit set
- If host does not announce MSS, it is assumed 536 (for non-local destinations). It can be also saved on per-route basis
- After determining MSS, host sends segments with DF set. If MTU is smaller on the path, ICMP is returned with next-hop MTU. If MTU is not included in ICMP message, IP stack must perform trial-and-error procedure to guess minimal MTU (may take few packets until MTU is guessed)
- Upon receiving ICMP error, CWND is not changed, but slow-start is initiated. As path can change, hosts try larger MTU (up to announced MSS) periodically – every 10 min
- *(G) ip icmp rate-limit unreachable [df] [<ms>] [log [<packets>] [<interval-ms>]]*
  ICMP "fragmentation needed but DF set" (3/4) messages are throttled one per 500 ms. It can be set independently for DF messages and all other ICMP messages
- Issues
  - PMTUD may not work if firewalls are on the path, which usually filter unreachables
  - Allow (ACL) unreachables
    - *permit icmp any any unreachable*
      permit icmp any any time-exceeded
  - Signall MSS
    - *(IF) ip tcp adjust-mss <value>*
      Better solution than clearing DF to allow fragmentation, is to signal MSS between endpoints. This is only for TCP traffic
  - Clear DF bit
    - Allow fragmentations by clearing DF bit with route map (should be usd as last resort)
    - *route-map Clear-DF permit 10*
      match ...
      set ip df 0
      - *interface <inbound if>*
        ip policy route-map Clear-DF

Switch MTU

- *(G) system mtu routing <bytes>*
  The system routing MTU is the maximum MTU for routed packets and is also the maximum MTU that the switch advertises in routing updates for protocols such as OSPF. Does not require a switch restart.
- *(G) system mtu jumbo <bytes>*
  Change the MTU size for all Gigabit Ethernet and 10-Gigabit Ethernet interfaces on the switch
- *(G) system mtu <bytes>*
  Change the MTU size for all Fast Ethernet interfaces

Tunnels

- IPSec is able to fragment and reassemble packets, GRE cannot do that (that's why DF is set)
- *(IF) tunnel path-mtu-discovery*
  External GRE IP header has DF always cleared, not coppied from original IP. This command causes DF to be coppied from original packet to GRE IP header.
- **1) GRE tunnel IP MTU is 1476 (1500 – 24 bytes for GRE header), DF not set**
  Packet 1500 is received. TCP segment is 1480, which is larger than GRE MTU 1476. Fragmentation takes place. 1st packet is 1456 (+20 IP), 2nd packet is 24 (+20 IP). Each packet is then encapsulated in GRE: 1st packet is 1500 (including 24 GRE), 2nd packet is 68 (including 24 GRE). Tunnel destination host removes GRE and forwards 2 independent IP packets to end station, which reassemble them.
- **2) GRE tunnel IP MTU is 1476 (1500 – 24 bytes for GRE header), DF set**
  Router receives 1500 with DF. Packet is dropped, and ICMP is sent back with MTU 1476 (from GRE tunnel endpoint). Packet is encapsulated with new MTU and sent
- **3) GRE tunnel IP MTU is 1476 (1500 – 24 bytes for GRE header), DF set or not, some smaller MTU between GRE endpoints, no tunnel PMTUD**
  Packet with 1476 is received. GRE is added, packet is sent as 1500. Intermediate link is 1400. Packet is fragmented (GRE header DF is 0), original IP is only in first fragment. Tunnel endpoint must reassembly those parts. Then GRE is removed and original packet is sent to end station
- **4) GRE tunnel IP MTU is 1476 (1500 – 24 bytes for GRE header), DF set, some smaller MTU between GRE endpoints, tunnel PMTUD enabled**
  Packet with 1476 is received. GRE is added and sent. Intermediate link drops packet (DF set) and sends ICMP (MTU 1400) to tunnel source (external IP header source). Router lowers tunnel MTU to 1376 (1400 – 24 GRE). As packet was dropped, host retransmits it with 1476, but this time router send ICMP to original host with new MTU 1376. Host uses new MTU
- **5) Pure IPSec tunnel mode, DF cleared**
  Packet 1500 is received. IPSec adds 52 bytes. Outgoing MTU is 1500 so packet is fragmented in a normal way
- **6) Pure IPSec tunnel mode, DF is set**
  IPSec always performs PMTUD. Encryption is always performed before fragmentation. Packet 1500 is received, 52 bytes are added by IPSec. Outgoing MTU is 1500 so packet is dropped and ICMP is sent back with MTU 1442 (1500 – 58, which is max IPSec header size). Now host sends 1442, IPSec adds 52, resulting in 1496. Now packet is sent, but intermediate links is 1400. ICMP is sent to IPSec router with MTU 1400, router lowers SA MTU to 1400. Now, when host re-sends packet with 1442, router drops and sends ICMP with MTU 1342 (1500 – 58 max IPSec header). Host now sends 1342, 52 is added, and packet is sent all the way.
- **7) GRE + IPSec**
  IPSec is usually in transport mode to carry GRE between endpoints, and GRE itself is encrypted. In transport mode we save 20 bytes. It is recommended to set ***ip mtu 1400*** on GRE tunnels to avoid double fragmentation

Routing

Basic Rules

- Route will not be installed in RIB if NH is unavailable
- Recursive lookup must point at the end at outgoing intf so L2 header can be built
- Route lookup: **1)** longest match => **2)** AD => **3)** lowest metric
- Route switching is used for faster lookup (process, fast switching, cef)

Administrative Distance

Directly connected	0
Static to interface/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

Admin Distance

- ***distance <distance> <ip> <mask> <acl>***
  Defined within a routing protocol (any), but is not protocol-specific. The ip/mask defines advertising router (source), and an acl defines which routes will get new distance
- If AD is manipulated, and two protocols have the same AD, the tie-breaker is the default, original AD for each protocol

Static routing

- ***(G) ip route <net> <mask> <gw> <AD>***
  Floating static route is used to provide backup route in case primary route disappears (primary must have lower AD than floating static)
- Static route to p2p WAN interfaces can be always used, as there is always only one receiver on the other end. Static route to LAN interface can be used only if there is a router in that LAN segment, with ***ip proxy arp*** enabled
- Static route to interface makes this network also „connected", so they can be advertised with ***network*** statements by some protocols. Only BGP and EIGRP are able to pick up such networks. Static to Null0 acts the same, as Null0 is an interface
- ***(G) ip route <net> <mask> [<interface>] dhcp***
  Add static route with NH acquired from DHCP. If more than one interface acquires IP from DHCP, use ***interface*** option, otherwise, first IP assigned will be used

ODR

- ***(G) router odr***
  Configured on hub only
- Hub router can automatically discover stub networks. Stub routers use a default route to the hub (also learned via ODR: ***0* 0.0.0.0 [160/1] via ...***)
- ODR carries only the network portion of the address, without a mask. Information is carried by CDP TLVs
- The metric (hop count) will never be more than 1
- Hello 60sec, Invalid 180sec – CDP timers are used. ODR advertisements stop if any other routing protocol is enabled on stub

PBR

- Policy Based Routing proceeds through route-map until match is found. If no match is found or match is found in route-map deny statement, the packet not dropped, but it is forwarded according to normal destination-based process
- ***(IF) ip policy route-map <name>***
  Affects incoming packets only
- ***(IF) ip route-cache policy***
  By default, PBR is process-switched unless CEF is enabled. Fast-switching is recommended if CEF is not enabled. It must be added before PBR is applied
- ***(IF) ip route-cache same-interface***
  May be required if next-hop points to the same interface (ex. NBMA)
- ***(RM) set ip next-hop verify-availability <ip>***
  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
- ***(RM) 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.
- ***(G) ip local policy route-map <name>***
  For traffic originated by the router. By default router-generated traffic does not pass any outbound ACLs.
- ***(RM) set ip default next-hop <ip>***
  Use default next hop if previous, configured next hops become unavailable

GRE

Len 1500

Original packet	IP	TCP/UDP	Data
	20B	20B	1460B

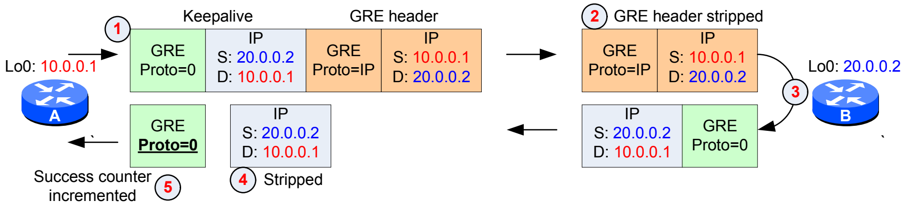
1500B ip mtu

Len 1500 / Len 1476

GRE IP	GRE	IP	TCP/UDP	Data
20B	4B	20B	20B	1436B

1500B ip mtu

- IP-in-IP
  - ***(IF) tunnel mode ipip***
  - Protocol 4 (IPv4)
  - not multiprotocol (no IPv6 inside)
- Protocol number 47. Multiprotocol, can carry CLNS, IPX, IPv6, etc
- Default bandwidth is 8kbp/s
- ***(IF) tunnel mode gre ip***
  Default mode, no need to specify
- ***(IF) tunnel route-via <if> {mandatory | prefered }***
  Tunnel route selection can be used, if there are multiple equal-cost paths to destination (only single route for tunnel destination is selected randomly). Mandatory: if there is no route via specified interface, tunnel goes down. Prefered: if there is no route via specified interface, tunnel takes next available path
- ***debug tunnel route-via***
- Recursive lookup error
  - Tunnel destination is learned through the tunnel itself
  - Tunnel goes down periodicaly
- Keepalive
  - ***(IF) keepalive <sec> <retry count>***
    By default keepalive is not enabled. No ability to bring down the line protocol, if the far end is unreachable
  - Keepalive works only for p2p GRE interfaces
  - If keepalive is enabled, NAT cannot be used for GRE packets



Lo0: 10.0.0.1
A
1
Keepalive
GRE header
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
GRE header 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
4
Stripped
IP S: 20.0.0.2 D: 10.0.0.1
GRE Proto=0
Success counter incremented
5

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 default to a permit and a sequence number of 10
- ***match ip address 10***
  match ip tag 2222
  Two different types of matches in the same route-map entry define **AND** operation (they all must match)
- ***match ip address 10 20***
  Two the same types of matches in the same route-map entry define **OR** operation (any of them can match)
- Continue
  - ***(RM) continue <seq>***
    Jump to specified seq or next seq if seq is not specified
  - If match clause exists, continue proceeds only if match is successful
  - 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
- Match
  - ***metric***: metric of the route (MED for BGP)
  - ***route-type***: OSPF or EIGRP route type (external, internal, type 1 or 2)
  - ***ip-address***: ACL defining specific prefix(es)
  - ***ip-address prefix-list***: specific prefix and length (bit netmask)
  - ***ip next-hop***: ACL defining route's next-hop (***via*** in routing table)
  - ***ip route-source***: ACL defining neighbor (***from*** in routing table)
  - ***tag***: route tag
  - ***length***: packet length

Routing

Default route

(G) ip default-gateway <ip>
Used not only on switches, but also on routers with ip routing disabled. When router is booting via TFTP, ip routing is not enabled yet, so this command may be needed.

(G) ip default-network <net>
Network must be in classful form and it must be in routing table. Makes that major network a candidate default. If you specify a subnet network (which must be in routing table also), IOS will automatically install major network as a static route with subnet network as a NH. The command with major network must be issued again to mark it as candicate default

To propagate default-network with EIGRP, this network must be coming from EIGRP. If it is defined as static, it must be either redistributed or advertised with network command

RIP will automatically advertise 0.0.0.0 if gateway of last resort is set with defaul-network

OSPF does not understand default-network at all

(G) ip route 0.0.0.0 0.0.0.0 <gw>
EIGRP and RIP can only propagate existing 0/0 via redistributing (for example, from static). OSPF does not understand 0/0 via redistribution unless ***default-information originate*** is added

IP Event Dampening

(IF) dampening [<half-life> <reuse> <suppress> <max> [restart]]
Reduce the effect of routing table instability. Mainly focused on IGP. Penatly is added (1000) every time interface flaps. Primary interface configuration is applied to all subinterfaces by default.

Half-life: Time, after which a penalty is decreased by half (default 5sec)

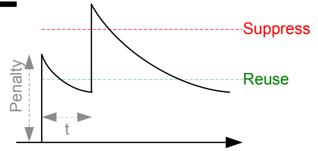
Reuse: When penalty decreases below this value, route is unsuppressed (default 1000)

Suppress: Suppress route when penalty is exceeded (default 2000)

Max: Maximum time a route can be suppressed (default 20 sec)

Restart: Penalty applied to interface when it comes up for the first time after reload (default 2000)

show interface dampening



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.

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

L1 adjacency detection

(IF) link debounce [time <msec>]
Available for switches. Default is 0 (disabled)

(IF) carrier-delay {msec <msec> | <sec>}
Available for routers. Default 2 sec. If carrier goes down, interface waits this long before communicating it

Advanced Object Tracking

(G) track <#> interface <if> {line-protocol | ip routing}
Go down when line-protocol goes down or interface loosed IP address (assigned by DHCP or IPCP)

(G) track <#> ip route <net>/<bits> {reachability | metric threshold}
Track route reachability or route's metric. Metric values are normalized to the range of 0 to 255, where 0 is connected and 255 is inaccessible. State is up if the scaled metric for that route is less than or equal to the *up* threshold. Tracking uses a per-protocol configurable resolution value to convert the real metric to the scaled metric

(G) track resolution ip route {eigrp | isis | ospf | static} <resolution-value>
Define resolutions for routes tracked with threshold. EIGRP resolution 256 - 40000000. ISIS resolution 1 - 1000. OSPF resolution 1 - 1562. Static resolution 1 to 100000

(G) track <#> ip sla <#> [state | reachability]
IP SLA tracking, in addition to up/down state, can set return codes

(G) track <#> list {boolean {and | or} | threshold {weight | percentage}}
List of tracked objects can be either ANDed or ORed. Objects can also be negated

(G) track <#> stub-object
Create dummy object that can be tracked and manipulated by EEM

(G) track timer {interface | ip route | sla } | list | stub}{<sec> | msec <msec>}
Defines interval during which the tracking process polls the tracked object. The default interval for interface polling is 1 sec, and for IP-route polling is 15 sec

default-state {up | down} - default state is up

(G) track 1 interface serial0/0 line-protocol
(G) track 2 interface serial0/1 line-protocol

```
track 12 list threshold weight
 object 1 weight 5
 object 2 weight 5
 threshold weight up 10 down 0
```
Object is down if two interfaces are down

Conditional 0/0 injection

```
track 1 sla 1 reachability
 delay down <sec> up <sec>
```
1. Track remote router with RTR

(G) ip route 192.0.0.192 255.255.255.255 null 0 track 1
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

(G) ip prefix-list TST permit 1.1.1.1/32
3. Create prefix-list covering bogus route and assign it to route-map

```
route-map TST permit 10
 match ip address prefix-list TST
```
4. Assign tracked prefix to route-map

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

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>*)

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

Chain distribution on one router is **NOT** possible. For example when redistributing EIGRP => RIP => OSPF, then EIGRP routes will be redistributed into RIP, but NOT into OSPF. Separate redistribution of EIGRP to OSPF needs to be configured

NSF/GR

Non Stop Forwarding is a way to continue forwarding packets while control plane is recovering from failure

Graceful Restart is a way of rebuilding forwarding data in routing protocols when control plane has recovered

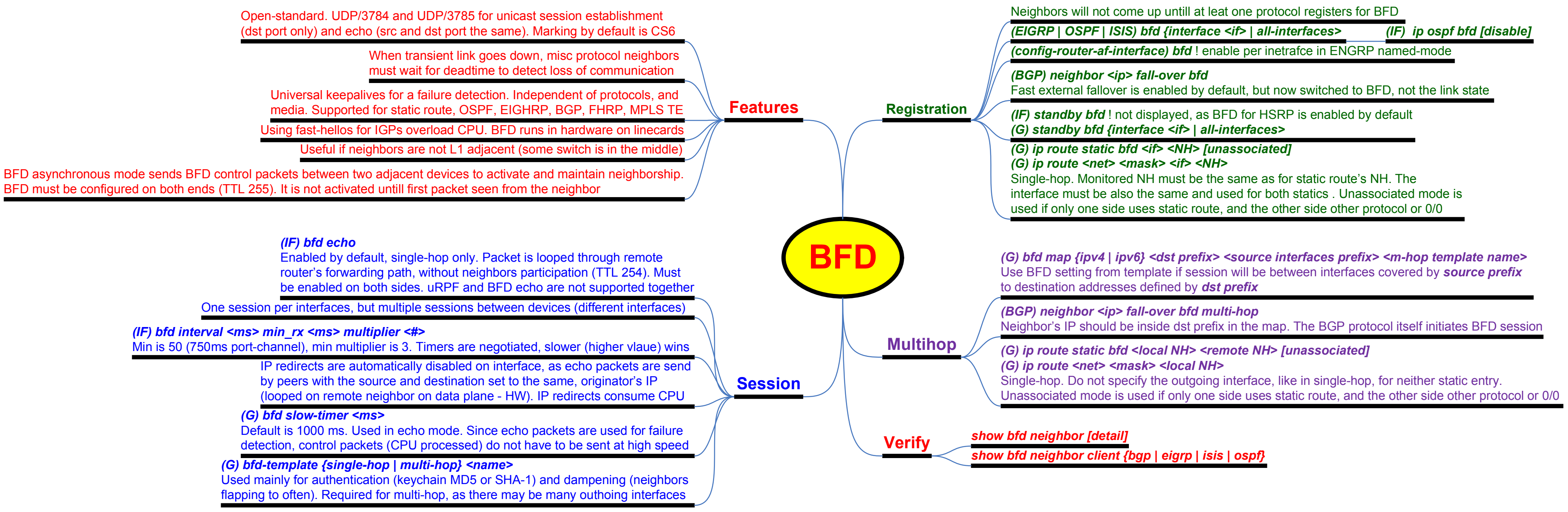
1) If NSF capable control place detects failure (neighbors down) it will not reset data plane, but will mark forwarding information as stale. Any traffic will be switched based on last known information

2) Control plane must recover before neighbor hold time expires. When control plane gets up, it signals the neighbor that it still forwards traffic, but would like to resync. This is GR message (protocol dependant)

3) Control plane must recover before neighbor hold time expires. When control plane gets up, it signals the neighbor that it still forwards traffic, but would like to resync

4) Neighbor then sends prefix updates. When done, end-of-table marker is sent

5) When end-of-table is seen, router recalculates topology and informs CEF, which removes stale entries



BFD

Features

Open-standard. UDP/3784 and UDP/3785 for unicast session establishment (dst port only) and echo (src and dst port the same). Marking by default is CS6

When transient link goes down, misc protocol neighbors must wait for deadtime to detect loss of communication

Universal keepalives for a failure detection. Independent of protocols, and media. Supported for static route, OSPF, EIGHRP, BGP, FHRP, MPLS TE

Using fast-hellos for IGPs overload CPU. BFD runs in hardware on linecards

Useful if neighbors are not L1 adjacent (some switch is in the middle)

BFD asynchronous mode sends BFD control packets between two adjacent devices to activate and maintain neighborship. BFD must be configured on both ends (TTL 255). It is not activated untill first packet seen from the neighbor

Session

(IF) bfd echo
Enabled by default, single-hop only. Packet is looped through remote router's forwarding path, without neighbors participation (TTL 254). Must be enabled on both sides. uRPF and BFD echo are not supported together

One session per interfaces, but multiple sessions between devices (different interfaces)

(IF) bfd interval <ms> min_rx <ms> multiplier <#>
Min is 50 (750ms port-channel), min multiplier is 3. Timers are negotiated, slower (higher vlaue) wins

IP redirects are automatically disabled on interface, as echo packets are send by peers with the source and destination set to the same, originator's IP (looped on remote neighbor on data plane - HW). IP redirects consume CPU

(G) bfd slow-timer <ms>
Default is 1000 ms. Used in echo mode. Since echo packets are used for failure detection, control packets (CPU processed) do not have to be sent at high speed

(G) bfd-template {single-hop | multi-hop} <name>
Used mainly for authentication (keychain MD5 or SHA-1) and dampening (neighbors flapping to often). Required for multi-hop, as there may be many outhoing interfaces

Registration

Neighbors will not come up untill at leat one protocol registers for BFD

(EIGRP | OSPF | ISIS) bfd {interface <if> | all-interfaces> ***(IF) ip ospf bfd [disable]***

(config-router-af-interface) bfd ! enable per inetrafce in ENGRP named-mode

(BGP) neighbor <ip> fall-over bfd
Fast external fallover is enabled by default, but now switched to BFD, not the link state

(IF) standby bfd ! not displayed, as BFD for HSRP is enabled by default
(G) standby bfd {interface <if> | all-interfaces>

(G) ip route static bfd <if> <NH> [unassociated]
(G) ip route <net> <mask> <if> <NH>
Single-hop. Monitored NH must be the same as for static route's NH. The interface must be also the same and used for both statics . Unassociated mode is used if only one side uses static route, and the other side other protocol or 0/0

Multihop

(G) bfd map {ipv4 | ipv6} <dst prefix> <source interfaces prefix> <m-hop template name>
Use BFD setting from template if session will be between interfaces covered by ***source prefix*** to destination addresses defined by ***dst prefix***

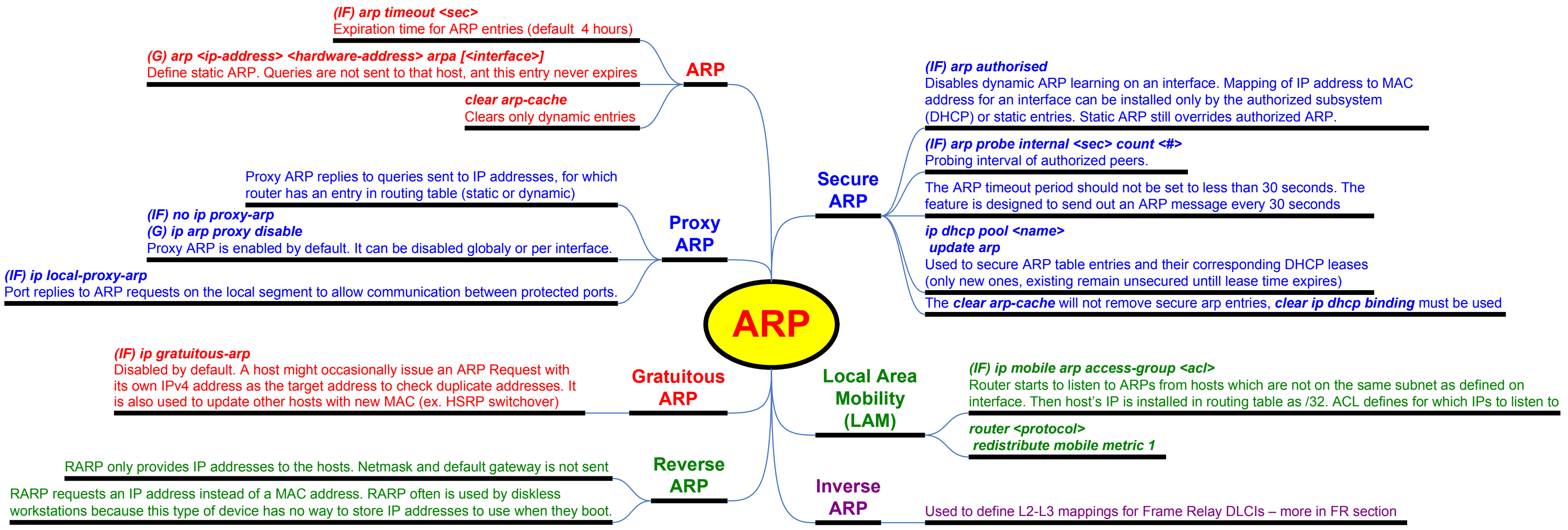
(BGP) neighbor <ip> fall-over bfd multi-hop
Neighbor's IP should be inside dst prefix in the map. The BGP protocol itself initiates BFD session

(G) ip route static bfd <local NH> <remote NH> [unassociated]
(G) ip route <net> <mask> <local NH>
Single-hop. Do not specify the outgoing interface, like in single-hop, for neither static entry. Unassociated mode is used if only one side uses static route, and the other side other protocol or 0/0

Verify

show bfd neighbor [detail]

show bfd neighbor client {bgp | eigrp | isis | ospf}



ARP

ARP

(IF) arp timeout <sec>
Expiration time for ARP entries (default 4 hours)

(G) arp <ip-address> <hardware-address> arpa [<interface>]
Define static ARP. Queries are not sent to that host, ant this entry never expires

clear arp-cache
Clears only dynamic entries

Proxy ARP

Proxy ARP replies to queries sent to IP addresses, for which router has an entry in routing table (static or dynamic)

(IF) no ip proxy-arp
(G) ip arp proxy disable
Proxy ARP is enabled by default. It can be disabled globaly or per interface.

(IF) ip local-proxy-arp
Port replies to ARP requests on the local segment to allow communication between protected ports.

Gratuitous ARP

(IF) ip gratuitous-arp
Disabled by default. A host might occasionally issue an ARP Request with its own IPv4 address as the target address to check duplicate addresses. It is also used to update other hosts with new MAC (ex. HSRP switchover)

Reverse ARP

RARP only provides IP addresses to the hosts. Netmask and default gateway is not sent

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.

Secure ARP

(IF) arp authorised
Disables dynamic ARP learning on an interface. Mapping of IP address to MAC address for an interface can be installed only by the authorized subsystem (DHCP) or static entries. Static ARP still overrides authorized ARP.

(IF) arp probe internal <sec> count <#>
Probing interval of authorized peers.

The ARP timeout period should not be set to less than 30 seconds. The feature is designed to send out an ARP message every 30 seconds

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

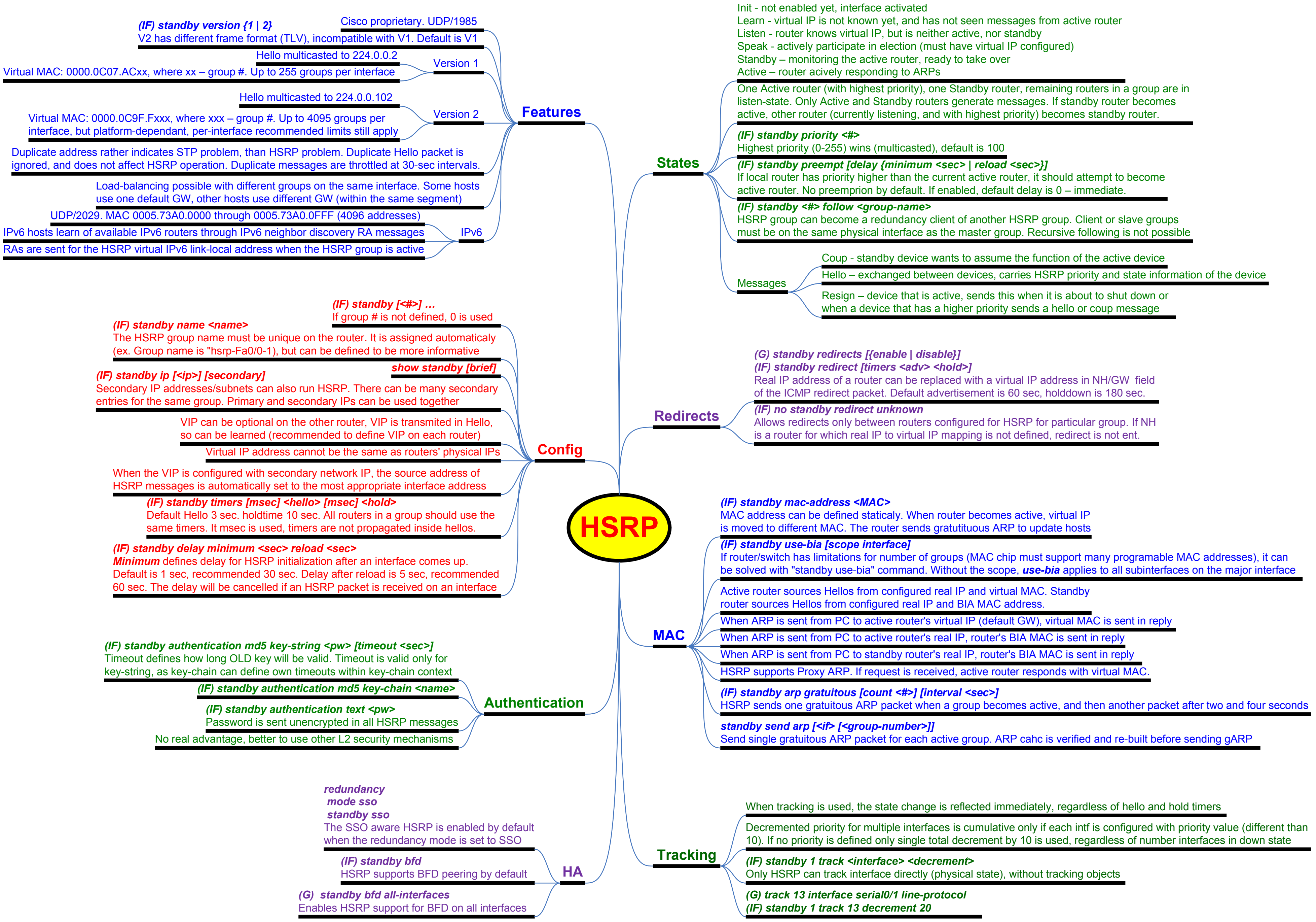
Local Area Mobility (LAM)

(IF) ip mobile arp access-group <acl>
Router starts to listen to ARPs from hosts which are not on the same subnet as defined 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

Inverse ARP

Used to define L2-L3 mappings for Frame Relay DLCIs – more in FR section



HSRP

Features

- Cisco proprietary. UDP/1985
- ***(IF) standby version {1 | 2}***
  V2 has different frame format (TLV), incompatible with V1. Default is V1
- Version 1
  - Hello multicasted to 224.0.0.2
  - Virtual MAC: 0000.0C07.ACxx, where xx – group #. Up to 255 groups per interface
- Version 2
  - Hello multicasted to 224.0.0.102
  - Virtual MAC: 0000.0C9F.Fxxx, where xxx – group #. Up to 4095 groups per interface, but platform-dependant, per-interface recommended limits still apply
- Duplicate address rather indicates STP problem, than HSRP problem. Duplicate Hello packet is ignored, and does not affect HSRP operation. Duplicate messages are throttled at 30-sec intervals.
- 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)
- IPv6
  - UDP/2029. MAC 0005.73A0.0000 through 0005.73A0.0FFF (4096 addresses)
  - IPv6 hosts learn of available IPv6 routers through IPv6 neighbor discovery RA messages
  - RAs are sent for the HSRP virtual IPv6 link-local address when the HSRP group is active

Config

- ***(IF) standby [<#>] …***
  If group # is not defined, 0 is used
- ***(IF) standby name <name>***
  The HSRP group name must be unique on the router. It is assigned automaticaly (ex. Group name is "hsrp-Fa0/0-1), but can be defined to be more informative
- ***show standby [brief]***
- ***(IF) standby ip [<ip>] [secondary]***
  Secondary IP addresses/subnets can also run HSRP. There can be many secondary entries for the same group. Primary and secondary IPs can be used together
- VIP can be optional on the other router, VIP is transmited in Hello, so can be learned (recommended to define VIP on each router)
- Virtual IP address cannot be the same as routers' physical IPs
- When the VIP is configured with secondary network IP, the source address of HSRP messages is automatically set to the most appropriate interface address
- ***(IF) standby timers [msec] <hello> [msec] <hold>***
  Default Hello 3 sec. holdtime 10 sec. All routers in a group should use the same timers. It msec is used, timers are not propagated inside hellos.
- ***(IF) standby delay minimum <sec> reload <sec>***
  Minimum defines delay for HSRP initialization after an interface comes up. Default is 1 sec, recommended 30 sec. Delay after reload is 5 sec, recommended 60 sec. The delay will be cancelled if an HSRP packet is received on an interface

Authentication

- ***(IF) standby authentication md5 key-string <pw> [timeout <sec>]***
  Timeout defines how long OLD key will be valid. Timeout is valid only for key-string, as key-chain can define own timeouts within key-chain context
- ***(IF) standby authentication md5 key-chain <name>***
- ***(IF) standby authentication text <pw>***
  Password is sent unencrypted in all HSRP messages
- No real advantage, better to use other L2 security mechanisms

HA

- ***redundancy***
  mode sso
  standby sso
  The SSO aware HSRP is enabled by default when the redundancy mode is set to SSO
- ***(IF) standby bfd***
  HSRP supports BFD peering by default
- ***(G) standby bfd all-interfaces***
  Enables HSRP support for BFD on all interfaces

States

- Init - not enabled yet, interface activated
  Learn - virtual IP is not known yet, and has not seen messages from active router
  Listen - router knows virtual IP, but is neither active, nor standby
  Speak - actively participate in election (must have virtual IP configured)
  Standby – monitoring the active router, ready to take over
  Active – router acively responding to ARPs
- One Active router (with highest priority), one Standby router, remaining routers in a group are in listen-state. Only Active and Standby routers generate messages. If standby router becomes active, other router (currently listening, and with highest priority) becomes standby router.
- ***(IF) standby priority <#>***
  Highest priority (0-255) wins (multicasted), default is 100
- ***(IF) standby preempt [delay {minimum <sec> | reload <sec>}]***
  If local router has priority higher than the current active router, it should attempt to become active router. No preemprion by default. If enabled, default delay is 0 – immediate.
- ***(IF) standby <#> follow <group-name>***
  HSRP group can become a redundancy client of another HSRP group. Client or slave groups must be on the same physical interface as the master group. Recursive following is not possible
- Messages
  - Coup - standby device wants to assume the function of the active device
  - Hello – exchanged between devices, carries HSRP priority and state information of the device
  - Resign – device that is active, sends this when it is about to shut down or when a device that has a higher priority sends a hello or coup message

Redirects

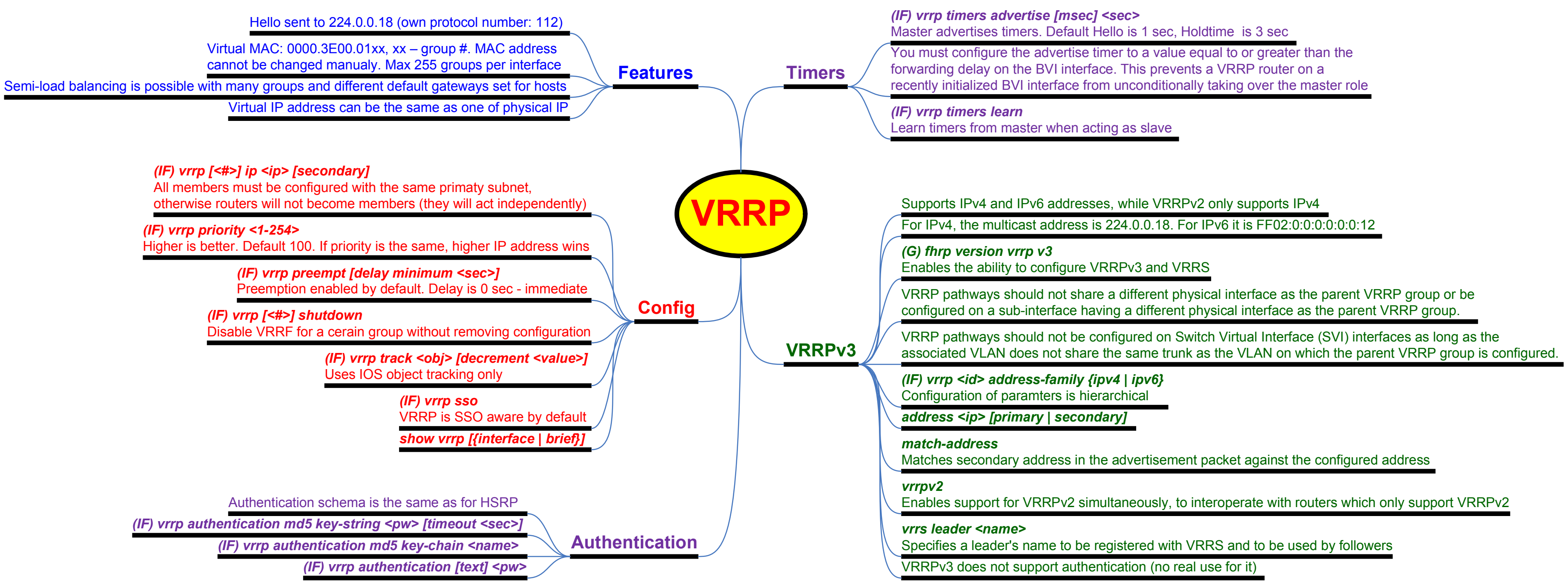
- ***(G) standby redirects [{enable | disable}]***
  (IF) standby redirect [timers <adv> <hold>]
  Real IP address of a router can be replaced with a virtual IP address in NH/GW field of the ICMP redirect packet. Default advertisement is 60 sec, holddown is 180 sec.
- ***(IF) no standby redirect unknown***
  Allows redirects only between routers configured for HSRP for particular group. If NH is a router for which real IP to virtual IP mapping is not defined, redirect is not ent.

MAC

- ***(IF) standby mac-address <MAC>***
  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
- ***(IF) standby use-bia [scope interface]***
  If router/switch has limitations for number of groups (MAC chip must support many programable MAC addresses), it can be solved with "standby use-bia" command. Without the scope, ***use-bia*** applies to all subinterfaces on the major interface
- Active router sources Hellos from configured real IP and virtual MAC. Standby router sources Hellos from configured real IP and BIA MAC address.
- When ARP is sent from PC to active router's virtual IP (default GW), virtual MAC is sent in reply
- When ARP is sent from PC to active router's real IP, router's BIA MAC is sent in reply
- When ARP is sent from PC to standby router's real IP, router's BIA MAC is sent in reply
- HSRP supports Proxy ARP. If request is received, active router responds with virtual MAC.
- ***(IF) standby arp gratuitous [count <#>] [interval <sec>]***
  HSRP sends one gratuitous ARP packet when a group becomes active, and then another packet after two and four seconds
- ***standby send arp [<if> [<group-number>]]***
  Send single gratuitous ARP packet for each active group. ARP cahc is verified and re-built before sending gARP

Tracking

- When tracking is used, the state change is reflected immediately, regardless of hello and hold timers
- 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
- ***(G) track 13 interface serial0/1 line-protocol***
  (IF) standby 1 track 13 decrement 20



VRRP

Features

- Hello sent to 224.0.0.18 (own protocol number: 112)
- Virtual MAC: 0000.3E00.01xx, xx – group #. MAC address cannot be changed manualy. Max 255 groups per interface
- Semi-load balancing is possible with many groups and different default gateways set for hosts
- Virtual IP address can be the same as one of physical IP

Timers

- ***(IF) vrrp timers advertise [msec] <sec>***
  Master advertises timers. Default Hello is 1 sec, Holdtime is 3 sec
- You must configure the advertise timer to a value equal to or greater than the forwarding delay on the BVI interface. This prevents a VRRP router on a recently initialized BVI interface from unconditionally taking over the master role
- ***(IF) vrrp timers learn***
  Learn timers from master when acting as slave

Config

- ***(IF) vrrp [<#>] ip <ip> [secondary]***
  All members must be configured with the same primaty subnet, otherwise routers will not become members (they will act independently)
- ***(IF) vrrp priority <1-254>***
  Higher is better. Default 100. If priority is the same, higher IP address wins
- ***(IF) vrrp preempt [delay minimum <sec>]***
  Preemption enabled by default. Delay is 0 sec - immediate
- ***(IF) vrrp [<#>] shutdown***
  Disable VRRF for a cerain group without removing configuration
- ***(IF) vrrp track <obj> [decrement <value>]***
  Uses IOS object tracking only
- ***(IF) vrrp sso***
  VRRP is SSO aware by default
- ***show vrrp [{interface | brief}]***

Authentication

- Authentication schema is the same as for HSRP
- ***(IF) vrrp authentication md5 key-string <pw> [timeout <sec>]***
- ***(IF) vrrp authentication md5 key-chain <name>***
- ***(IF) vrrp authentication [text] <pw>***

VRRPv3

- Supports IPv4 and IPv6 addresses, while VRRPv2 only supports IPv4
- For IPv4, the multicast address is 224.0.0.18. For IPv6 it is FF02:0:0:0:0:0:0:12
- ***(G) fhrp version vrrp v3***
  Enables the ability to configure VRRPv3 and VRRS
- VRRP pathways should not share a different physical interface as the parent VRRP group or be configured on a sub-interface having a different physical interface as the parent VRRP group.
- VRRP pathways should not be configured on Switch Virtual Interface (SVI) interfaces as long as the associated VLAN does not share the same trunk as the VLAN on which the parent VRRP group is configured.
- ***(IF) vrrp <id> address-family {ipv4 | ipv6}***
  Configuration of paramters is hierarchical
- ***address <ip> [primary | secondary]***
- ***match-address***
  Matches secondary address in the advertisement packet against the configured address
- ***vrrpv2***
  Enables support for VRRPv2 simultaneously, to interoperate with routers which only support VRRPv2
- ***vrrs leader <name>***
  Specifies a leader's name to be registered with VRRS and to be used by followers
- VRRPv3 does not support authentication (no real use for it)

GLBP

Features

- Cisco proprietary. Hello multicasted to 224.0.0.102, UDP/3222
- AVG assigns unique MAC to each router: 0007.B400.xxyy, xx – group #, yy – router #
- One primary AVG, one backup AVG, other members in a group sre in listening state. If primary fails, one of AVF with highest priority/IP (backup AVG) is elected to be primary AVG. Other routers in listening state can become primary AVF
- Up to 4 primary forwarders in a group. They have MAC addresses assigned by AVG in a sequence. Other routers in a group are secondary forwarders in listening state – they learn virtual MACs via Hello
- If AVF fails, other AVF awainting in listening stae, becomes primary AVF. The AVG starts two timers for failed AVF, redirect and timeout

Timers

- ***(IF) glbp timers [msec] <hello> [msec] <hold>***
  Default Hello 3 sec. Holdtime 10 sec. Sub-second hello can be configured
- ***(IF) glbp timers redirect <redirect> <timeout>***
  redirect – during this time, AVG keeps redirecting hosts to that AVF
  timeout – after this time, AVF is removed from all gateways in a group, AVG stops pointing ARPs to that AVF, but AVF keeps forwarding existing traffic

Authentication

- Authentication schema is the same as for HSRP
- ***(IF) glbp authentication text <pw>***
- ***(IF) glbp authentication md5 key-chain <name>***
- ***(IF) glbp authentication md5 key-string <pw>***

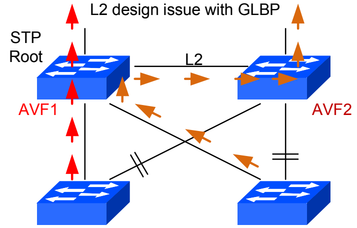
Config

- ***(IF) glbp [<#>] …***
  Max 1024 GLBP groups per physical interface. Default group is 0 (not shown in config)
- ***(IF) glbp priority <1-255>***
  Higher priority is better (default 100). If priority is the same, higher IP address wins
- ***(IF) glbp ip [<ip> [secondary]]***
  IP has to be defined on AVG. GLBP can also run for secondary addresses
- ***(IF) glbp client-cache maximum <#> [timeout <sec>]***
  AVG keeps client cache containing which AVF is assigned to which host. Max 2000 hosts. If max is reached, oldest entries are removed. Timeout defined how long entries are kept in cache (without ARP query from a client). Recommended timeout – little longer that hots ARP cache timeout
- ***(IF) glbp preempt [delay minimum <sec>]***
  No AVG preemption by default. Delay can be defined before preemption takes place
- ***(IF) glbp forwarder preempt [delay minimum <sec>]***
  Backup AVF can become active AVF if weighting drops below low threshold for 30 sec. This feature is enabled by default
- ***show glbp [{brief | detail}]***

True Load balancing

- ***(IF) glbp weighting track <id> [decrement <value>]***
  (IF) glbp 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
- ***(IF) glbp load-balancing {host-dependent | weighted | round-robin}***
  Define load-balancing method. AVG by default responds to hosts' ARP with virtual MAC requests in round-robin fashion
- Host-dependent load balancing is required by SNAT. Not recommended for small number of hosts. Given host is guaranteed to use the same MAC
- ***RT1: glbp 1 weighting 20***
  RT2: glbp 1 weighting 10
  In weighted mode each router advertises weighting and assignements. Weighted load-balancing in ratio 2:1

L2 design issue with GLBP



FHRP

IRDP

- ICMP Router Discovery Protocol. Uses ICMP messages to advertise candidate default gateway. By default messages are broadcasted
- Each device discovered becomes a candidate for the default router, and a new highest-priority router is selected when a higher priority router is discovered, when the current default router is declared down, or when a TCP connection is about to time out because of excessive retransmissions
- ***(IF) ip irdp address <ip> <preference>***
  Advertises IP address configured on interface as a gateway. Optionaly, different IPs (many) can be advertised with different priorities (all defined IPs are advertised)
- Advertisements vary between ***minadvertinterval*** and ***maxadvertinterval***
- Server
  - ***(IF) ip irdp***
  - ***(IF) ip irdp multicast*** (enable mutlicasting to 224.0.0.1)
  - ***(IF) ip irdp holdtime <sec>*** (default is 30 min)
  - ***(IF) ip irdp maxadvertinterval <sec>*** (default is 450 sec)
  - ***(IF) ip irdp minadvertinterval <sec>*** (default is 600 sec)
  - ***(IF) ip irdp preference <#>*** (default is 0; higher is better)
- Client
  - ***(G) no ip routing***
  - ***(G) ip gdp irdp***

DRP

- It enables the Cisco Distributed 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>***

PFR

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 expands capabilities that facilitate intelligent route control on a per application basis
- OER can learn both outside and inside prefixes.
- Master controller and Border Router can be enabled on the same router

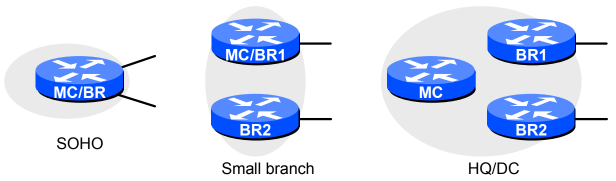
Master Controller

Features

- Monitors the network and maintains a central policy database with statistics. 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
- 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

- ***(G) oer master***
  Enable OER master controller. Below commands are defined in its context
  - ***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>***
    Dynamic port used for communication between MC and BR. Must be the same on both sides
  - ***logging***
    Enables syslog messages for a master controller (*notice* level)
  - ***keepalive <sec>***
    Keepalive between MC and BR. Default is 60 sec.



SOHO

Small branch

HQ/DC

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
- ***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

- ***(G) oer border***
  Enable OER border router
  - ***port <port>***
    Port used between MC and BR
  - ***local <intf>***
    Identifies source for communication with an OER MC
  - ***master <ip> key-chain <name>***
    Define MC. Key chain is mandatory

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)

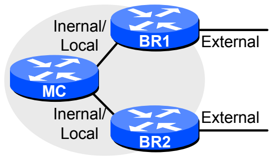
- Routing can be manipulated with artificialy injected more-specific routes. Measured prefixes' parent route (the same or wider prefix) with a valid next hop must exist for prefix to be injected
- In control mode commands are sent back to the border routers to alter routing in the OER managed network 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>

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

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]***

PFR Measure

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

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 to gather performance statistics of current WAN link
- 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]***

Fast probe

- ***oer master***
  mode monitor both
  Active and Passive 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.
- Uses IPSLA to monitor all other links to determine possible alternate exit

PFR Learn

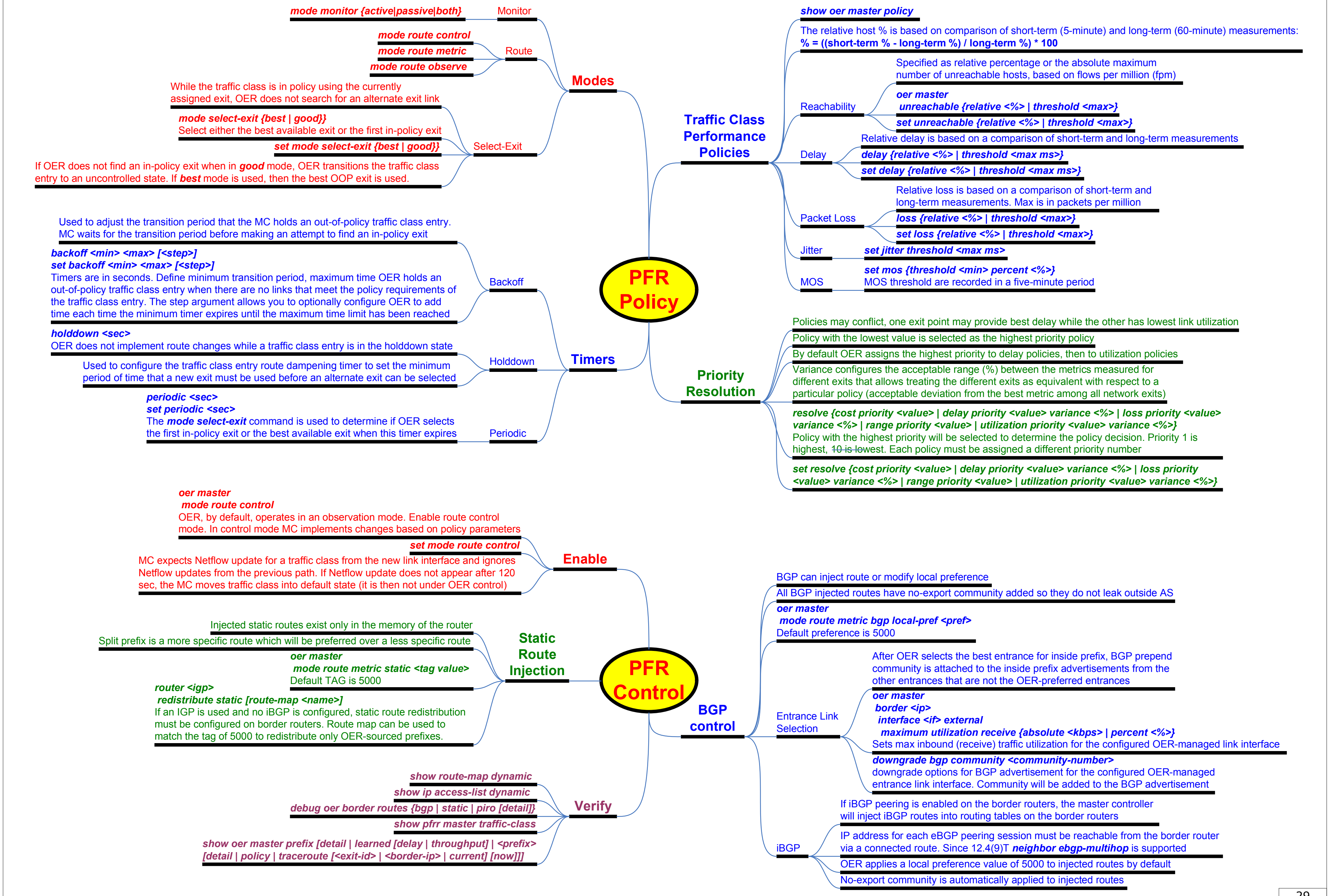
Automatic learning (*learn*)

- ***(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

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



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
  - *holddown <sec>*
    OER does not implement route changes while a traffic class entry is in the holddown state
  - 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
- 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 lo~~west. 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 <%>}*

PFR 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 pfrr 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

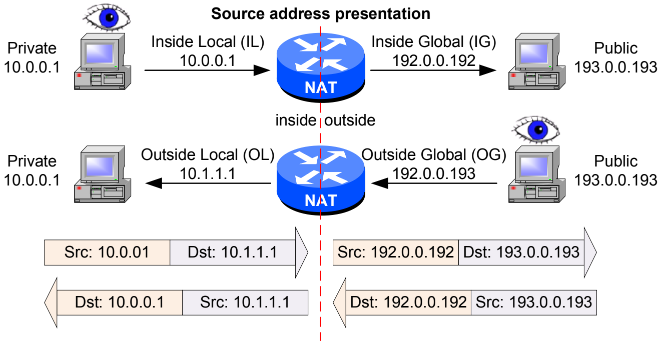
Inside-to-Outside

- if IPSec then check input access list
- decryption
- input access list (again, if IPSec)
- input rate limits
- input accounting
- redirect to web cache
- policy routing
- routing
- **NAT inside to outside**
- crypto (mark for encryption)
- output access list
- inspect (CBAC)
- TCP intercept
- encryption
- queueing

Outside-to-Inside

- If IPSec then check input access list
- decryption
- input access list
- input rate limits
- input accounting
- redirect to web cache
- **NAT outside to inside**
- policy routing
- routing
- crypto (mark for encryption)
- output access list
- inspect (CBAC)
- TCP intercept
- encryption
- queueing

Source address presentation



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
- ***(IF) ip nat {inside | outside}*** - Define interface role for NAT
- If router does not have a route to destination, packet is unroutable, and does not use NAT. This can be also a case when ***no ip classless*** is configured
- If a translation entry already exists and matches traffic then it this entry will be used, and neither access lists nor route map will be consulted
- 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
- ***(G) ip nat inside {source | destination} ...***
  (G) ip nat outside source ...
  Inside and outside define on which interface traffic arrives when performing NAT. Source and destination define which address is to be translated
- Route-map can be used when doing source (only) translation to define more granular policy
- 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

Dynamic

- Dynamic NAT is considered a security feature, as there cannot be a traffic flowing from outside to inside untill the NAT entry is present which is initiated from inside to outside
- **PAT**
  - ***(G) ip nat inside source list <acl> interface <if> 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
- ***(IF) ip nat pool <name> <start> <end> {netmask <mask> | prefix-length <prefix>} [type match-host]***
  match-host: host portion of the IG will match the host portion of the IL. Netmask defines the range of addresses for which the router listens (is aware) when packets arrive, so it knows what should be sent to NAT engine
- ***(G) ip nat inside source list <acl> pool <name>***
  Translate dynamicaly source addresses of inside hosts. Make sure ACL does not catch control traffic (EIGRP,...)
- When IG or OL addresses belong to directly attached interface, router created ***ip aliases***, so it can answer ARP requests. If there is no NAT entry for such address, and router runs specific service, it can be attacked – router answers to packets (ICMP or UDP) not realy destined for it

Static

- ***(G) 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.
- ***(G) ip nat inside source static network <local net> <global net> <mask or prefix len>***
  Network translation assigns last octed one-to-one
- ***(G) ip nat inside source static tcp 192.168.1.1 21 192.1.1.3 21 extendable***
  (G) ip nat inside source static tcp 192.168.1.3 80 192.1.1.3 80 extendable
  Statically mapping an IG address to more than one IL address is not allowed. To allow service distribution ***extendable*** keyword must be used. This is only for incoming traffic from outside. Outgoing traffic falls under dynamic NAT. If it's not configured, traffic is dropped
- ***(G) ip nat inside source static tcp <IL> <port> <IG> <port> [no-alias]***
  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.
- ***(G) ip nat inside source static <IL> <IG> redundancy <name>***
  Redundancy with HRP. Active router is performing NAT translation

Stateful

- ***(G) ip nat inside source list <acl> pool <name> mapping <mapping id>***
- With HSRP
  - ***ip nat stateful id <id>***
    redundancy <HSRP name>
    mapping-id <id>
    Mapping-id identifies translations and must be the same on both routers. Stateful-id must be unique on each router
- Without 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***

Verify

- ***show ip nat translation***
- ***show ip nat statistics***
- ***clear ip nat translation ****
- NAT translation failure codes (***debug ip nat***)
  - A = Inside to outside fails after routing
    B = Outside to inside fails before routing
    C = Outside to inside fails after routing
    D = Helpered fails
    L = Internally generated packet fails
    E = Inside to outside fails after routing

NAT

NVI

Cisco recommends that you use legacy NAT for VRF to global NAT (ip nat inside/out) and between interfaces in the same VRF. NVI is used for NAT between different VRFs.

NVI0 interface is created

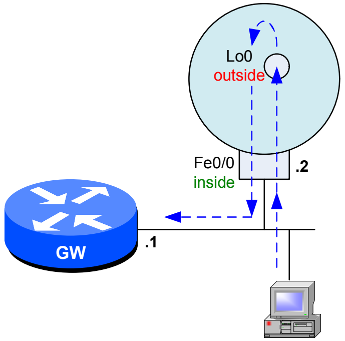
(IF) ip nat enable
NVI removes the requirements to configure an interface as either NAT inside or NAT outside

(IF) ip nat {source | destination} ...
No need to specify inside and outside in translation definitions

show ip nat nvi {translations | statistics}

NAT on a stick

If you have ISP modem on the same network and a router with single interface

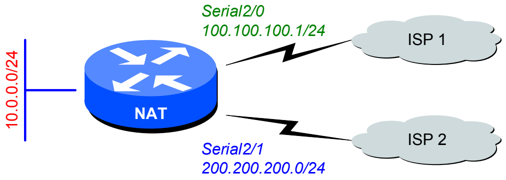


```
interface Loopback0
 ip address 10.1.1.1 255.255.255.252
 ip nat outside

access-list NAT permit ...

route-map RM-NAT permit 10
 match ip address NAT
 set ip next-hop 10.1.1.2

interface FastEthernet0/0
 ip address 192.168.1.2 255.255.255.0
 ip nat inside
 ip policy route-map RM-NAT

ip route 0.0.0.0 0.0.0.0 192.168.1.1
```

Multihoming to 2 ISPs

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

```
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
```



Virtual reassembly

Router tracks fragments and delays them (holds) until all fragments are received or reassembly timeout expires (then incomplete packet is dropped). It is “virtual” reassembly, as packet is not put back into one, but only stored localy for NAT processing, after which, all fragments are sent to destination

(IF) ip virtual-reassembly [max-reassemblies <#>] [max-fragments <#>] [timeout <sec>] [drop-fragments]

max-reassembies – defines max simultaneous packets to be tracked. Drops packets if max is reached
max-fragments – max number of fragments for single packet (exceeding will be dropped)
timeout – how long router will wait for all fragments before dropping whole incomplete packet
drop-fragments – drop all fragments arriving on interface

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>

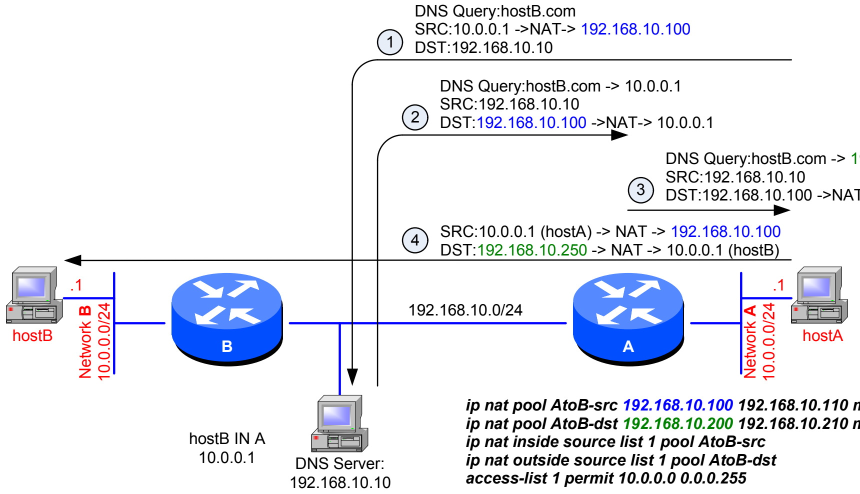
2. Associate global IP (single IPs), by which local servers are seen from outside
ip nat inside destination list <acl> pool <name>
access-list <acl> permit <global IP>

(G) 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

Overlapping networks

DNS can be used to allow overlaping networks to communicate. Returning reply from DNS server is translated (DNS payload information) with ***ip nat outside source*** command

If DNS is not used then static translation has to be used (ip nat outside source static), but it is more difficult to manage



1
DNS Query:hostB.com
SRC:10.0.0.1 ->NAT-> 192.168.10.100
DST:192.168.10.10
2
DNS Query:hostB.com -> 10.0.0.1
SRC:192.168.10.10
DST:192.168.10.100 ->NAT-> 10.0.0.1
3
DNS Query:hostB.com -> 192.168.10.250
SRC:192.168.10.10
DST:192.168.10.100 ->NAT-> 10.0.0.1
4
SRC:10.0.0.1 (hostA) -> NAT -> 192.168.10.100
DST:192.168.10.250 -> NAT -> 10.0.0.1 (hostB)
hostB
.1
Network B
10.0.0.0/24
B
192.168.10.0/24
A
Network A
10.0.0.0/24
.1
hostA
hostB IN A
10.0.0.1
DNS Server:
192.168.10.10

```
ip nat pool AtoB-src 192.168.10.100 192.168.10.110 mask 255.255.255.0
ip nat pool AtoB-dst 192.168.10.200 192.168.10.210 mask 255.255.255.0
ip nat inside source list 1 pool AtoB-src
ip nat outside source list 1 pool AtoB-dst
access-list 1 permit 10.0.0.0 0.0.0.255
```

DHCP

Features

(G) service dhcp (enabled by default)

UDP/67 server; UDP/68 client; Payload is 300 bytes

Client has fixed UDP/68 port as reply is broadcasted to the segment and if random port was used other hosts would receive „unknown" packets. Here, they know it is a BOOTP reply.

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

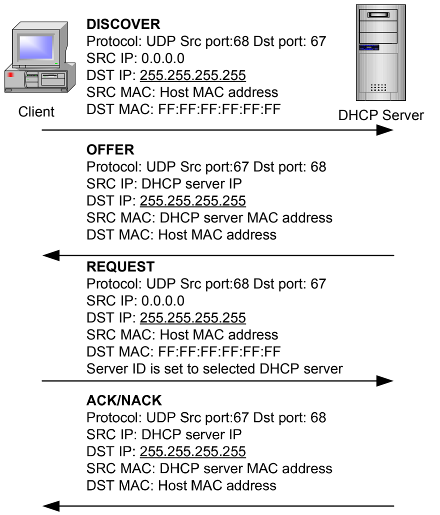
Address is assigned with lease time. Client can extend lease time dynamically sending DHCPREQUEST, usualu at 50% of time. If server sends DKCPACK, lease is extended. If server sends DHCPNACK, client restarts the full lease. If no response is received, client uses an address until lease expires

Transaction ID (random) field is used to distinguish different queries. „Seconds" field can be used by secondary server not to respond until this time expires and reply is not heard from primary server

When server replies, it places static arp entry in local cache for a client's MAC and assigned IP, so ARP request does not have to be generated, otherwise client could not respond to that ARP request as it doesn't know own IP yet (chicken and egg)

Cisco IOS DHCP server can allocate IP based on the relay information option (option 82) information sent by the relay agent. In some networks, it is necessary to use additional information to further determine which IP addresses to allocate

show ip dhcp binding



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

Oper. 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)			
Vendor-specific options (64B)			

Client

(IF) ip address dhcp
Assign IP address from DHCP. When 0/0 is also defined in the pool, the router install static 0/0

(IF) ip dhcp client request ...
Request additional parameters (options)

(IF) ip dhcp client lease <days> [<hours>]
Request specific lease time for an address

(IF) ip address dhcp client-id <if>
Specify Client-ID to identify specific profile on DHCP server. Client ID and MAC address are two different fields

(#) {release | renew} dhcp <if>
Force interface to release and renew IP address

Authentication

Authentication mechanism allows servers to determine whether a request for DHCP information comes from a client that is authorized to use the network

When FORCERENEW request is authenticated, client renews its lease according to normal DHCP procedures, otherwise request is dropped

(IF) ip dhcp client authentication key-chain <name>

(IF) ip dhcp client authentication mode md5

(EXEC) ip dhcp-client forcerenew

Relay

(IF) ip helper address <ip> [redundancy <HSRP name>]
Broadcast is changed to directed unicast with router's LAN interface's IP address as a source (source and destination NAT is performed). 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

Dynamic Binding

(G) ip dhcp exclude-address <start> <end>
Multiple lines defining which addresses in a network range will not be assigned to clients

(G) ip dhcp database flash:/bindings [timeout <sec>] [write-delay <sec>]
Configure database agent for storing bindings, and conflict logging

(G) 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)

ip dhcp pool <name>
 network <net> [<mask>] [secondary]
 default-router <ip> (max 8)
 dns-server <ip> (max 8)
 domain-name <name>
 lease <days> [<hours>]
 option <id> <type> <value> (additional options – ex. 150 TFTP server, etc) →
 netbios-name-server <<ip>> (max 8)
 netbios-node-type <type> (h-node: Hybrid node recommended)
 utilization mark {high | low} <%> [log]
 bootfile <filename>
 option <code> [instance <#>] {ascii <string> | hex <string> | <ip-address>}
 accounting <aaa method>

Subnet mask	1
Router (gateways)	3
DNS servers	6
Hostname	12
Domain name	15
Static routes	33
WINS server	44
NetBIOS node type	46
Lease time	51
Message type	53
Server identifier	54
Renewal time	58
Rebinding time	59
Unique identifier	61
TFTP Server	150

(G) ip dhcp ping {packets <#> | timeout <msec>}
DHCP server pings IP before it is leased (default 2 sec). It also sweep-pings whole range when pool is defined

show ip dhcp {pool | binding | conflict | database}

(G) ip dhcp bootp ignore
Ignore BOOTP requests sent to this DHCP server

DHCP server can respond to a BOOTP request, but it may not be desired. The BOOTP server is usually configured with static bindings for the BOOTP clients.

Static Binding

ip dhcp pool PC1
 host <ip> /24
 hardware-address <MAC>
 client-identifier <id>
Host pools inherit entire configuration from the main pool (IP is matched against network in the pool). When creating per-host pool, 01 must be added in the front of MAC defined as client-id (01 means ethernet media type). Ex. **01**00.0c12.213e.23. Some DHCP clients send a client identifier (DHCP option 61) in the DHCP packet. It must be configured to allow assignment.

ip dhcp pool <name>
 origin file <url >
Static Mapping feature enables assignment of static IPs without creating many individual host pools

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

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.

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

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

Secondary Pool

Router looks for a free address in the primary subnet. When the primary subnet is exhausted, the DHCP server automatically looks for a free address in any secondary subnets

If the giaddr matches a secondary subnet in the pool, the DHCP server allocates an IP address from that secondary subnet (even if IP addresses are available in the primary subnet

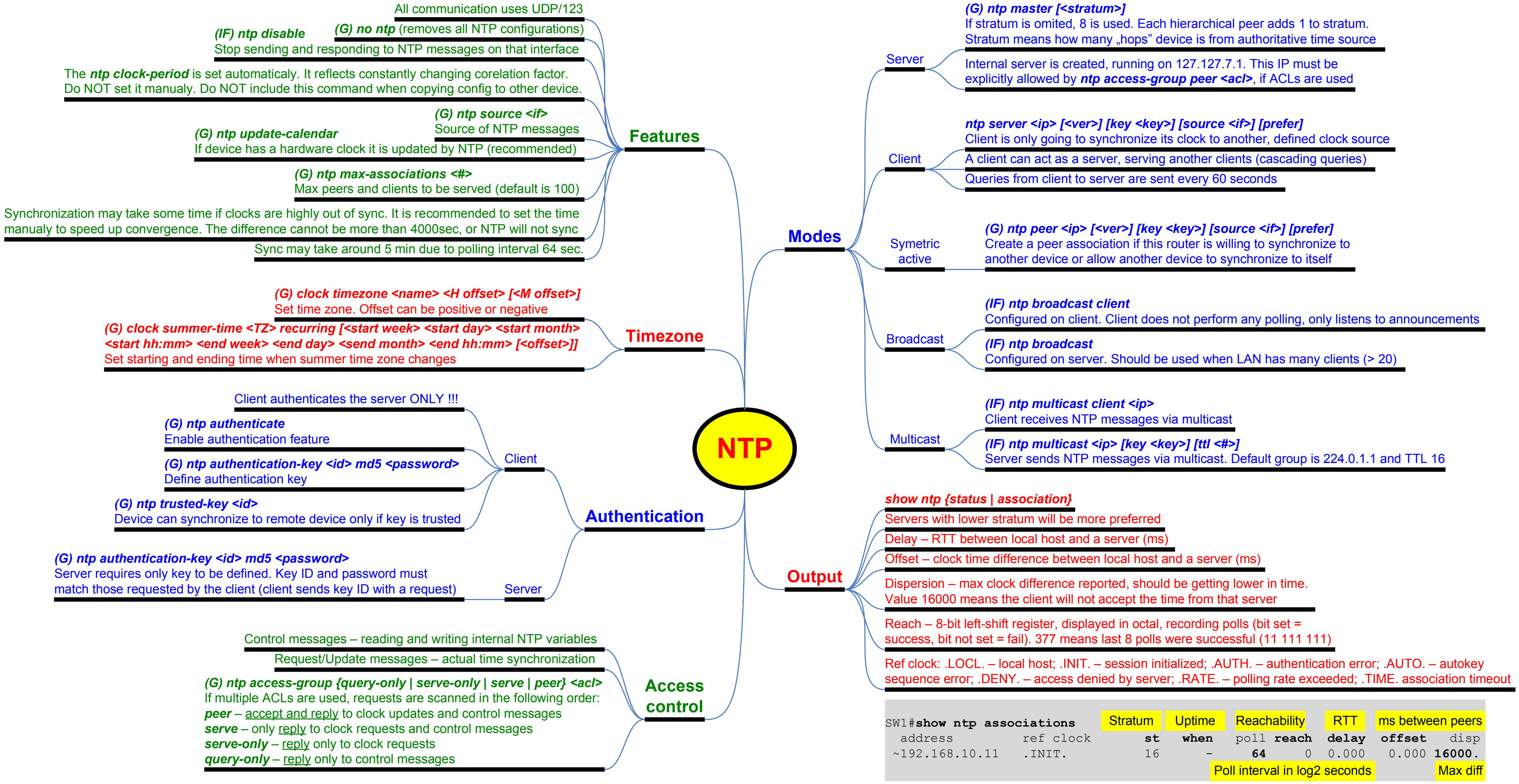
network <net> [<mask>] secondary
 override default-router <ip> (max 8)
 override utilization mark {high | low} <%> [log]

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

(G) ip address-pool dhcp-proxy-client
(G) ip dhcp-server <ip>



NTP

Features

- All communication uses UDP/123
- *(G) no ntp* (removes all NTP configurations)
- *(IF) ntp disable*
  Stop sending and responding to NTP messages on that interface
- The *ntp clock-period* is set automaticaly. It reflects constantly changing corelation factor.
  Do NOT set it manualy. Do NOT include this command when copying config to other device.
- *(G) ntp source <if>*
  Source of NTP messages
- *(G) ntp update-calendar*
  If device has a hardware clock it is updated by NTP (recommended)
- *(G) ntp max-associations <#>*
  Max peers and clients to be served (default is 100)
- Synchronization may take some time if clocks are highly out of sync. It is recommended to set the time manualy to speed up convergence. The difference cannot be more than 4000sec, or NTP will not sync
- Sync may take around 5 min due to polling interval 64 sec.

Timezone

- *(G) clock timezone <name> <H offset> [<M offset>]*
  Set time zone. Offset can be positive or negative
- *(G) clock summer-time <TZ> recurring [<start week> <start day> <start month> <start hh:mm> <end week> <end day> <send month> <end hh:mm> [<offset>]]*
  Set starting and ending time when summer time zone changes

Authentication

- Client
  - Client authenticates the server ONLY !!!
  - *(G) ntp authenticate*
    Enable authentication feature
  - *(G) ntp authentication-key <id> md5 <password>*
    Define authentication key
  - *(G) ntp trusted-key <id>*
    Device can synchronize to remote device only if key is trusted
- Server
  - *(G) ntp authentication-key <id> md5 <password>*
    Server requires only key to be defined. Key ID and password must match those requested by the client (client sends key ID with a request)

Access control

- Control messages – reading and writing internal NTP variables
- Request/Update messages – actual time synchronization
- *(G) 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

Modes

- Server
  - *(G) ntp master [<stratum>]*
    If stratum is omited, 8 is used. Each hierarchical peer adds 1 to stratum.
    Stratum means how many „hops" device is from authoritative time source
  - 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
  - *ntp server <ip> [<ver>] [key <key>] [source <if>] [prefer]*
    Client is only going to synchronize its clock to another, defined clock source
  - A client can act as a server, serving another clients (cascading queries)
  - Queries from client to server are sent every 60 seconds
- Symetric active
  - *(G) ntp peer <ip> [<ver>] [key <key>] [source <if>] [prefer]*
    Create a peer association if this router is willing to synchronize to another device or allow another device to synchronize to itself
- Broadcast
  - *(IF) ntp broadcast client*
    Configured on client. Client does not perform any polling, only listens to announcements
  - *(IF) ntp broadcast*
    Configured on server. Should be used when LAN has many clients (> 20)
- Multicast
  - *(IF) ntp multicast client <ip>*
    Client receives NTP messages via multicast
  - *(IF) ntp multicast <ip> [key <key>] [ttl <#>]*
    Server sends NTP messages via multicast. Default group is 224.0.1.1 and TTL 16

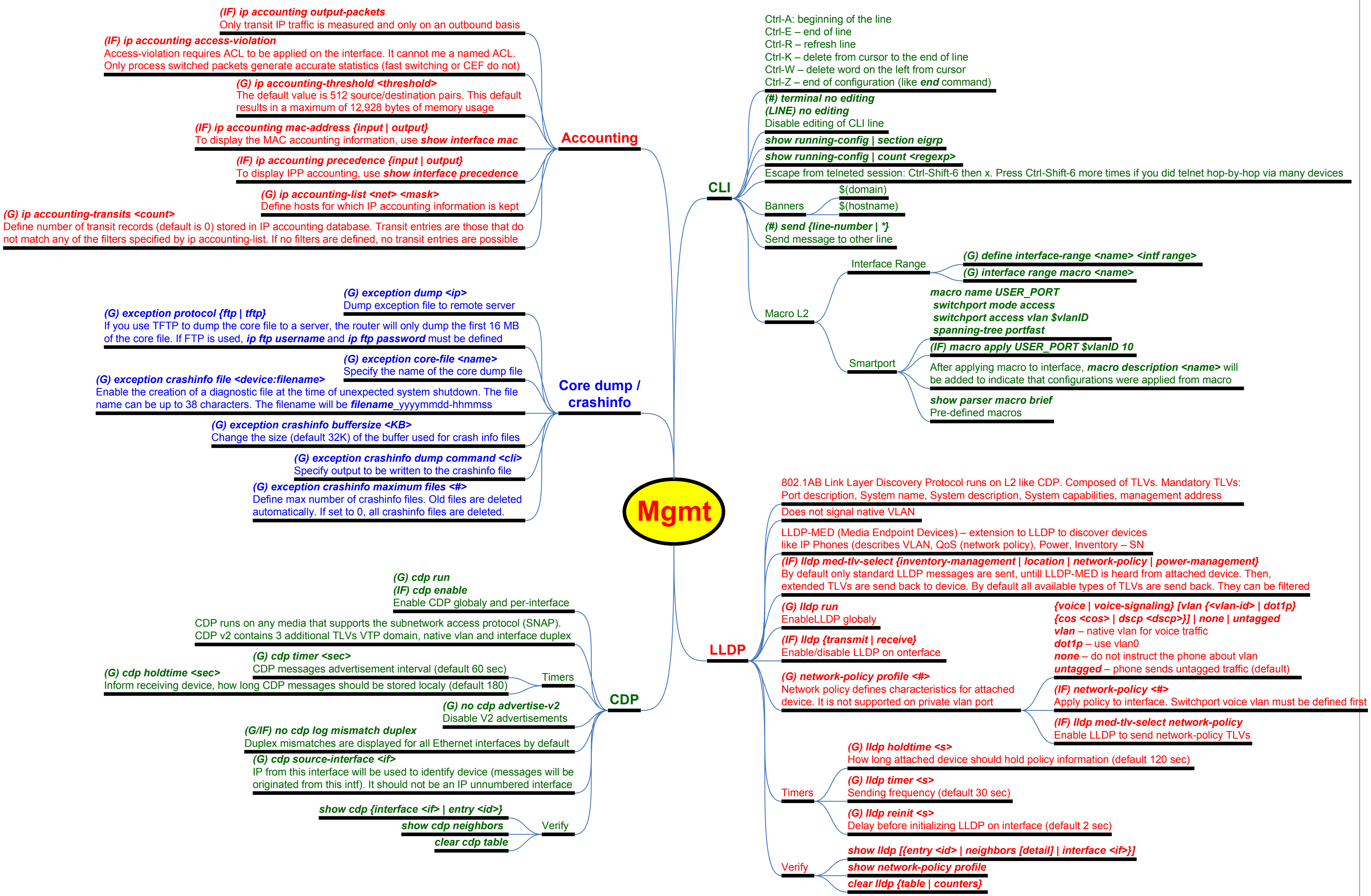
Output

- *show ntp {status | association}*
- Servers with lower stratum will be more preferred
- Delay – RTT between local host and a server (ms)
- Offset – clock time difference between local host and a server (ms)
- Dispersion – max clock difference reported, should be getting lower in time.
  Value 16000 means the client will not accept the time from that server
- Reach – 8-bit left-shift register, displayed in octal, recording polls (bit set = success, bit not set = fail). 377 means last 8 polls were successful (11 111 111)
- Ref clock: .LOCL. – local host; .INIT. – session initialized; .AUTH. – authentication error; .AUTO. – autokey sequence error; .DENY. – access denied by server; .RATE. – polling rate exceeded; .TIME. association timeout

Stratum | Uptime | Reachability | RTT | ms between peers

```
SW1#show ntp associations
  address           ref clock          st   when   poll reach  delay  offset   disp
 ~192.168.10.11    .INIT.              16      -     64     0  0.000   0.000 16000.
```

Poll interval in log2 seconds | Max diff



Mgmt

Accounting

- ***(IF) ip accounting output-packets***
  Only transit IP traffic is measured and only on an outbound basis
- ***(IF) ip accounting access-violation***
  Access-violation requires ACL to be applied on the interface. It cannot me a named ACL.
  Only process switched packets generate accurate statistics (fast switching or CEF do not)
- ***(G) 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
- ***(IF) ip accounting mac-address {input | output}***
  To display the MAC accounting information, use ***show interface mac***
- ***(IF) ip accounting precedence {input | output}***
  To display IPP accounting, use ***show interface precedence***
- ***(G) ip accounting-list <net> <mask>***
  Define hosts for which IP accounting information is kept
- ***(G) ip accounting-transits <count>***
  Define number of transit records (default is 0) stored in IP accounting database. Transit entries are those that do not match any of the filters specified by ip accounting-list. If no filters are defined, no transit entries are possible

Core dump / crashinfo

- ***(G) exception dump <ip>***
  Dump exception file to remote server
- ***(G) exception protocol {ftp | tftp}***
  If you use TFTP to dump the core file to a server, the router will only dump the first 16 MB of the core file. If FTP is used, ***ip ftp username*** and ***ip ftp password*** must be defined
- ***(G) exception core-file <name>***
  Specify the name of the core dump file
- ***(G) exception crashinfo file <device:filename>***
  Enable the creation of a diagnostic file at the time of unexpected system shutdown. The file name can be up to 38 characters. The filename will be ***filename***_yyyymmdd-hhmmss
- ***(G) exception crashinfo buffersize <KB>***
  Change the size (default 32K) of the buffer used for crash info files
- ***(G) exception crashinfo dump command <cli>***
  Specify output to be written to the crashinfo file
- ***(G) exception crashinfo maximum files <#>***
  Define max number of crashinfo files. Old files are deleted automatically. If set to 0, all crashinfo files are deleted.

CDP

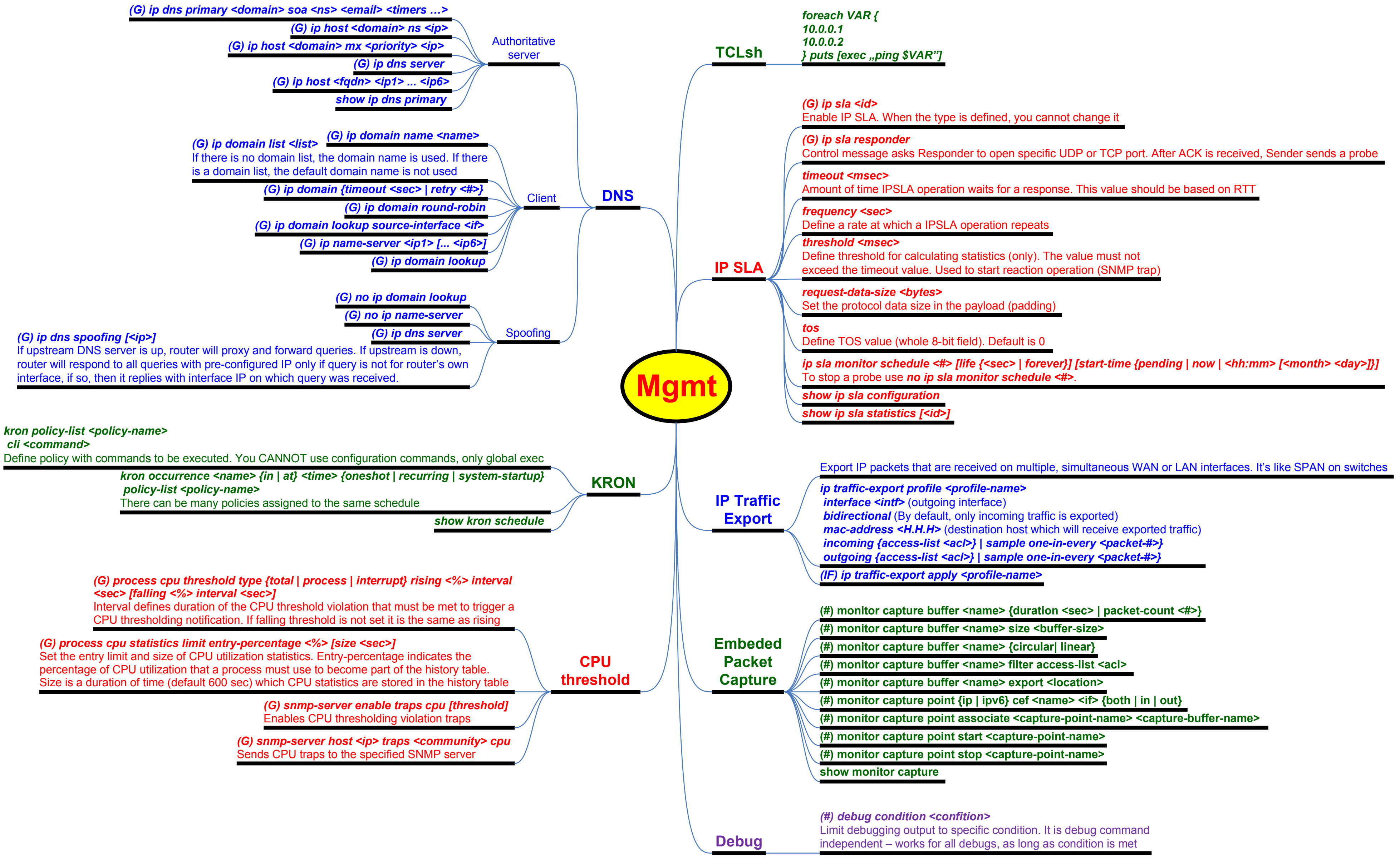
- ***(G) cdp run***
  (IF) cdp enable
  Enable CDP globaly and per-interface
- CDP runs on any media that supports the subnetwork access protocol (SNAP).
  CDP v2 contains 3 additional TLVs VTP domain, native vlan and interface duplex
- Timers
  - ***(G) cdp timer <sec>***
    CDP messages advertisement interval (default 60 sec)
  - ***(G) cdp holdtime <sec>***
    Inform receiving device, how long CDP messages should be stored localy (default 180)
- ***(G) no cdp advertise-v2***
  Disable V2 advertisements
- ***(G/IF) no cdp log mismatch duplex***
  Duplex mismatches are displayed for all Ethernet interfaces by default
- ***(G) 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
- Verify
  - ***show cdp {interface <if> | entry <id>}***
  - ***show cdp neighbors***
  - ***clear cdp table***

CLI

- Ctrl-A: beginning of the line
  Ctrl-E – end of line
  Ctrl-R – refresh line
  Ctrl-K – delete from cursor to the end of line
  Ctrl-W – delete word on the left from cursor
  Ctrl-Z – end of configuration (like ***end*** command)
- ***(#) terminal no editing***
  (LINE) no editing
  Disable editing of CLI line
- ***show running-config | section eigrp***
- ***show running-config | count <regexp>***
- Escape from telneted session: Ctrl-Shift-6 then x. Press Ctrl-Shift-6 more times if you did telnet hop-by-hop via many devices
- Banners
  - $(domain)
  - $(hostname)
- ***(#) send {line-number | *}***
  Send message to other line
- Macro L2
  - Interface Range
    - ***(G) define interface-range <name> <intf range>***
    - ***(G) interface range macro <name>***
  - Smartport
    - ***macro name USER_PORT***
      switchport mode access
      switchport access vlan $vlanID
      spanning-tree portfast
    - ***(IF) macro apply USER_PORT $vlanID 10***
    - After applying macro to interface, ***macro description <name>*** will be added to indicate that configurations were applied from macro
    - ***show parser macro brief***
      Pre-defined macros

LLDP

- 802.1AB Link Layer Discovery Protocol runs on L2 like CDP. Composed of TLVs. Mandatory TLVs: Port description, System name, System description, System capabilities, management address
- Does not signal native VLAN
- LLDP-MED (Media Endpoint Devices) – extension to LLDP to discover devices like IP Phones (describes VLAN, QoS (network policy), Power, Inventory – SN
- ***(IF) lldp med-tlv-select {inventory-management | location | network-policy | power-management}***
  By default only standard LLDP messages are sent, untill LLDP-MED is heard from attached device. Then, extended TLVs are send back to device. By default all available types of TLVs are send back. They can be filtered
- ***(G) lldp run***
  EnableLLDP globaly
- ***(IF) lldp {transmit | receive}***
  Enable/disable LLDP on onterface
- ***(G) network-policy profile <#>***
  Network policy defines characteristics for attached device. It is not supported on private vlan port
  - ***{voice | voice-signaling} [vlan {<vlan-id> | dot1p} {cos <cos> | dscp <dscp>}] | none | untagged}***
    vlan – native vlan for voice traffic
    dot1p – use vlan0
    none – do not instruct the phone about vlan
    untagged – phone sends untagged traffic (default)
  - ***(IF) network-policy <#>***
    Apply policy to interface. Switchport voice vlan must be defined first
  - ***(IF) lldp med-tlv-select network-policy***
    Enable LLDP to send network-policy TLVs
- Timers
  - ***(G) lldp holdtime <s>***
    How long attached device should hold policy information (default 120 sec)
  - ***(G) lldp timer <s>***
    Sending frequency (default 30 sec)
  - ***(G) lldp reinit <s>***
    Delay before initializing LLDP on interface (default 2 sec)
- Verify
  - ***show lldp [{entry <id> | neighbors [detail] | interface <if>}]***
  - ***show network-policy profile***
  - ***clear lldp {table | counters}***



Mgmt

DNS

Authoritative server

(G) ip dns primary <domain> soa <ns> <email> <timers …>

(G) ip host <domain> ns <ip>

(G) ip host <domain> mx <priority> <ip>

(G) ip dns server

(G) ip host <fqdn> <ip1> ... <ip6>

show ip dns primary

Client

(G) ip domain name <name>

(G) ip domain list <list>
If there is no domain list, the domain name is used. If there is a domain list, the default domain name is not used

(G) ip domain {timeout <sec> | retry <#>}

(G) ip domain round-robin

(G) ip domain lookup source-interface <if>

(G) ip name-server <ip1> [... <ip6>]

(G) ip domain lookup

Spoofing

(G) no ip domain lookup

(G) no ip name-server

(G) ip dns server

(G) 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, if so, then it replies with interface IP on which query was received.

KRON

kron policy-list <policy-name>
cli <command>
Define policy with commands to be executed. You CANNOT use configuration commands, only global exec

kron occurrence <name> {in | at} <time> {oneshot | recurring | system-startup}
policy-list <policy-name>
There can be many policies assigned to the same schedule

show kron schedule

CPU threshold

(G) process cpu threshold type {total | process | interrupt} rising <%> interval <sec> [falling <%> interval <sec>]
Interval defines duration of the CPU threshold violation that must be met to trigger a CPU thresholding notification. If falling threshold is not set it is the same as rising

(G) process cpu statistics limit entry-percentage <%> [size <sec>]
Set the entry limit and size of CPU utilization statistics. Entry-percentage indicates the percentage of CPU utilization that a process must use to become part of the history table. Size is a duration of time (default 600 sec) which CPU statistics are stored in the history table

(G) snmp-server enable traps cpu [threshold]
Enables CPU thresholding violation traps

(G) snmp-server host <ip> traps <community> cpu
Sends CPU traps to the specified SNMP server

TCLsh

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

IP SLA

(G) ip sla <id>
Enable IP SLA. When the type is defined, you cannot change it

(G) ip sla responder
Control message asks Responder to open specific UDP or TCP port. After ACK is received, Sender sends a probe

timeout <msec>
Amount of time IPSLA operation waits for a response. This value should be based on RTT

frequency <sec>
Define a rate at which a IPSLA operation repeats

threshold <msec>
Define threshold for calculating statistics (only). The value must not exceed the timeout value. Used to start reaction operation (SNMP trap)

request-data-size <bytes>
Set the protocol data size in the payload (padding)

tos
Define TOS value (whole 8-bit field). Default is 0

ip sla monitor schedule <#> [life {<sec> | forever}] [start-time {pending | now | <hh:mm> [<month> <day>]}]
To stop a probe use ***no ip sla monitor schedule <#>***.

show ip sla configuration

show ip sla statistics [<id>]

IP Traffic Export

Export IP packets that are received on multiple, simultaneous WAN or LAN interfaces. It's like SPAN on switches

ip traffic-export profile <profile-name>
interface <intf> (outgoing interface)
bidirectional (By default, only incoming traffic is exported)
mac-address <H.H.H> (destination host which will receive exported traffic)
incoming {access-list <acl>} | sample one-in-every <packet-#>}
outgoing {access-list <acl>} | sample one-in-every <packet-#>}

(IF) ip traffic-export apply <profile-name>

Embeded Packet Capture

(#) monitor capture buffer <name> {duration <sec> | packet-count <#>}

(#) monitor capture buffer <name> size <buffer-size>

(#) monitor capture buffer <name> {circular| linear}

(#) monitor capture buffer <name> filter access-list <acl>

(#) monitor capture buffer <name> export <location>

(#) monitor capture point {ip | ipv6} cef <name> <if> {both | in | out}

(#) monitor capture point associate <capture-point-name> <capture-buffer-name>

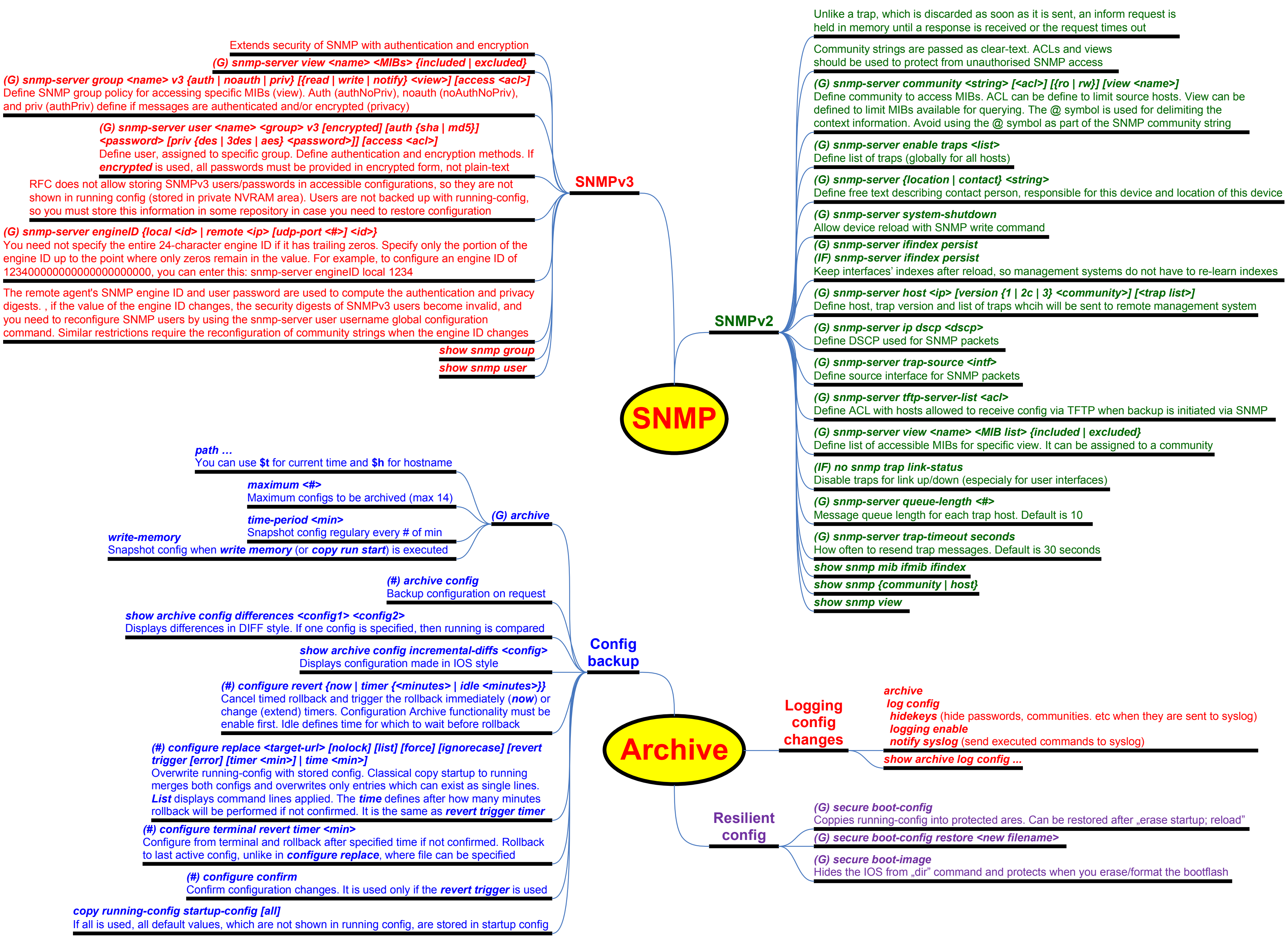
(#) monitor capture point start <capture-point-name>

(#) monitor capture point stop <capture-point-name>

show monitor capture

Debug

(#) debug condition <confition>
Limit debugging output to specific condition. It is debug command independent – works for all debugs, as long as condition is met



SNMP

SNMPv3

Extends security of SNMP with authentication and encryption

(G) snmp-server view <name> <MIBs> {included | excluded}

(G) snmp-server group <name> v3 {auth | noauth | priv} [{read | write | notify} <view>] [access <acl>]
Define SNMP group policy for accessing specific MIBs (view). Auth (authNoPriv), noauth (noAuthNoPriv), and priv (authPriv) define if messages are authenticated and/or encrypted (privacy)

(G) snmp-server user <name> <group> v3 [encrypted] [auth {sha | md5}] <password> [priv {des | 3des | aes} <password>]] [access <acl>]
Define user, assigned to specific group. Define authentication and encryption methods. If ***encrypted*** is used, all passwords must be provided in encrypted form, not plain-text

RFC does not allow storing SNMPv3 users/passwords in accessible configurations, so they are not shown in running config (stored in private NVRAM area). Users are not backed up with running-config, so you must store this information in some repository in case you need to restore configuration

(G) snmp-server engineID {local <id> | remote <ip> [udp-port <#>] <id>}
You need not specify the entire 24-character engine ID if it has trailing zeros. Specify only the portion of the engine ID up to the point where only zeros remain in the value. For example, to configure an engine ID of 123400000000000000000000, you can enter this: snmp-server engineID local 1234

The remote agent's SNMP engine ID and user password are used to compute the authentication and privacy digests. , if the value of the engine ID changes, the security digests of SNMPv3 users become invalid, and you need to reconfigure SNMP users by using the snmp-server user username global configuration command. Similar restrictions require the reconfiguration of community strings when the engine ID changes

show snmp group

show snmp user

SNMPv2

Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out

Community strings are passed as clear-text. ACLs and views should be used to protect from unauthorised SNMP access

(G) snmp-server community <string> [<acl>] [{ro | rw}] [view <name>]
Define community to access MIBs. ACL can be define to limit source hosts. View can be defined to limit MIBs available for querying. The @ symbol is used for delimiting the context information. Avoid using the @ symbol as part of the SNMP community string

(G) snmp-server enable traps <list>
Define list of traps (globally for all hosts)

(G) snmp-server {location | contact} <string>
Define free text describing contact person, responsible for this device and location of this device

(G) snmp-server system-shutdown
Allow device reload with SNMP write command

(G) snmp-server ifindex persist
(IF) snmp-server ifindex persist
Keep interfaces' indexes after reload, so management systems do not have to re-learn indexes

(G) snmp-server host <ip> [version {1 | 2c | 3} <community>] [<trap list>]
Define host, trap version and list of traps whcih will be sent to remote management system

(G) snmp-server ip dscp <dscp>
Define DSCP used for SNMP packets

(G) snmp-server trap-source <intf>
Define source interface for SNMP packets

(G) snmp-server tftp-server-list <acl>
Define ACL with hosts allowed to receive config via TFTP when backup is initiated via SNMP

(G) snmp-server view <name> <MIB list> {included | excluded}
Define list of accessible MIBs for specific view. It can be assigned to a community

(IF) no snmp trap link-status
Disable traps for link up/down (especialy for user interfaces)

(G) snmp-server queue-length <#>
Message queue length for each trap host. Default is 10

(G) snmp-server trap-timeout seconds
How often to resend trap messages. Default is 30 seconds

show snmp mib ifindex

show snmp {community | host}

show snmp view

Archive

Config backup

(G) archive

path …
You can use **$t** for current time and **$h** for hostname

maximum <#>
Maximum configs to be archived (max 14)

time-period <min>
Snapshot config regulary every # of min

write-memory
Snapshot config when ***write memory*** (or ***copy run start***) is executed

(#) archive config
Backup configuration on request

show archive config differences <config1> <config2>
Displays differences in DIFF style. If one config is specified, then running is compared

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

(#) configure revert {now | timer {<minutes> | idle <minutes>}}
Cancel timed rollback and trigger the rollback immediately (***now***) or change (extend) timers. Configuration Archive functionality must be enable first. Idle defines time for which to wait before rollback

(#) configure replace <target-url> [nolock] [list] [force] [ignorecase] [revert trigger [error] [timer <min>] | time <min>]
Overwrite running-config with stored config. Classical copy startup to running merges both configs and overwrites only entries which can exist as single lines. ***List*** displays command lines applied. The ***time*** defines after how many minutes rollback will be performed if not confirmed. It is the same as ***revert trigger timer***

(#) configure terminal revert timer <min>
Configure from terminal and rollback after specified time if not confirmed. Rollback to last active config, unlike in ***configure replace***, where file can be specified

(#) configure confirm
Confirm configuration changes. It is used only if the ***revert trigger*** is used

copy running-config startup-config [all]
If all is used, all default values, which are not shown in running config, are stored in startup config

Logging config changes

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

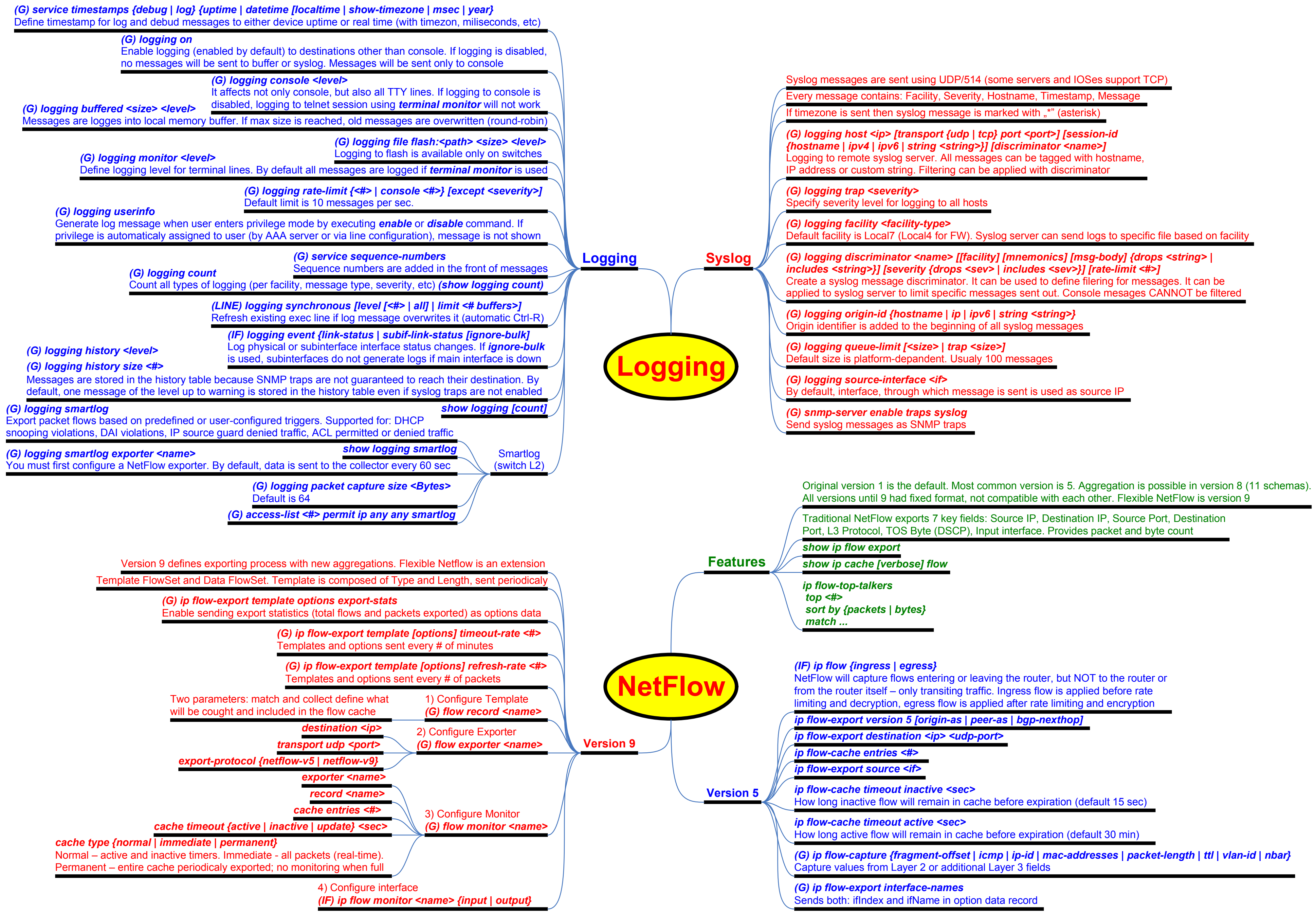
show archive log config ...

Resilient config

(G) secure boot-config
Coppies running-config into protected ares. Can be restored after „erase startup; reload"

(G) secure boot-config restore <new filename>

(G) secure boot-image
Hides the IOS from „dir" command and protects when you erase/format the bootflash



Logging

Logging

- ***(G) service timestamps {debug | log} {uptime | datetime [localtime | show-timezone | msec | year}***
  Define timestamp for log and debud messages to either device uptime or real time (with timezon, miliseconds, etc)
- ***(G) logging on***
  Enable logging (enabled by default) to destinations other than console. If logging is disabled, no messages will be sent to buffer or syslog. Messages will be sent only to console
- ***(G) logging console <level>***
  It affects not only console, but also all TTY lines. If logging to console is disabled, logging to telnet session using ***terminal monitor*** will not work
- ***(G) logging buffered <size> <level>***
  Messages are logges into local memory buffer. If max size is reached, old messages are overwritten (round-robin)
- ***(G) logging file flash:<path> <size> <level>***
  Logging to flash is available only on switches
- ***(G) logging monitor <level>***
  Define logging level for terminal lines. By default all messages are logged if ***terminal monitor*** is used
- ***(G) logging rate-limit {<#> | console <#>} [except <severity>]***
  Default limit is 10 messages per sec.
- ***(G) logging userinfo***
  Generate log message when user enters privilege mode by executing ***enable*** or ***disable*** command. If privilege is automaticaly assigned to user (by AAA server or via line configuration), message is not shown
- ***(G) service sequence-numbers***
  Sequence numbers are added in the front of messages
- ***(G) logging count***
  Count all types of logging (per facility, message type, severity, etc) ***(show logging count)***
- ***(LINE) logging synchronous [level [<#> | all] | limit <# buffers>]***
  Refresh existing exec line if log message overwrites it (automatic Ctrl-R)
- ***(IF) logging event {link-status | subif-link-status [ignore-bulk]***
  Log physical or subinterface interface status changes. If ***ignore-bulk*** is used, subinterfaces do not generate logs if main interface is down
- ***(G) logging history <level>***
  (G) logging history size <#>
  Messages are stored in the history table because SNMP traps are not guaranteed to reach their destination. By default, one message of the level up to warning is stored in the history table even if syslog traps are not enabled
- ***show logging [count]***
- Smartlog (switch L2)
  - ***(G) logging smartlog***
    Export packet flows based on predefined or user-configured triggers. Supported for: DHCP snooping violations, DAI violations, IP source guard denied traffic, ACL permitted or denied traffic
  - ***show logging smartlog***
  - ***(G) logging smartlog exporter <name>***
    You must first configure a NetFlow exporter. By default, data is sent to the collector every 60 sec
  - ***(G) logging packet capture size <Bytes>***
    Default is 64
  - ***(G) access-list <#> permit ip any any smartlog***

Syslog

- Syslog messages are sent using UDP/514 (some servers and IOSes support TCP)
- Every message contains: Facility, Severity, Hostname, Timestamp, Message
- If timezone is sent then syslog message is marked with „*" (asterisk)
- ***(G) logging host <ip> [transport {udp | tcp} port <port>] [session-id {hostname | ipv4 | ipv6 | string <string>}] [discriminator <name>]***
  Logging to remote syslog server. All messages can be tagged with hostname, IP address or custom string. Filtering can be applied with discriminator
- ***(G) logging trap <severity>***
  Specify severity level for logging to all hosts
- ***(G) logging facility <facility-type>***
  Default facility is Local7 (Local4 for FW). Syslog server can send logs to specific file based on facility
- ***(G) logging discriminator <name> [[facility] [mnemonics] [msg-body] {drops <string> | includes <string>}] [severity {drops <sev> | includes <sev>}] [rate-limit <#>]***
  Create a syslog message discriminator. It can be used to define filering for messages. It can be applied to syslog server to limit specific messages sent out. Console messages CANNOT be filtered
- ***(G) logging origin-id {hostname | ip | ipv6 | string <string>}***
  Origin identifier is added to the beginning of all syslog messages
- ***(G) logging queue-limit [<size> | trap <size>]***
  Default size is platform-dependant. Usualy 100 messages
- ***(G) logging source-interface <if>***
  By default, interface, through which message is sent is used as source IP
- ***(G) snmp-server enable traps syslog***
  Send syslog messages as SNMP traps

NetFlow

Features

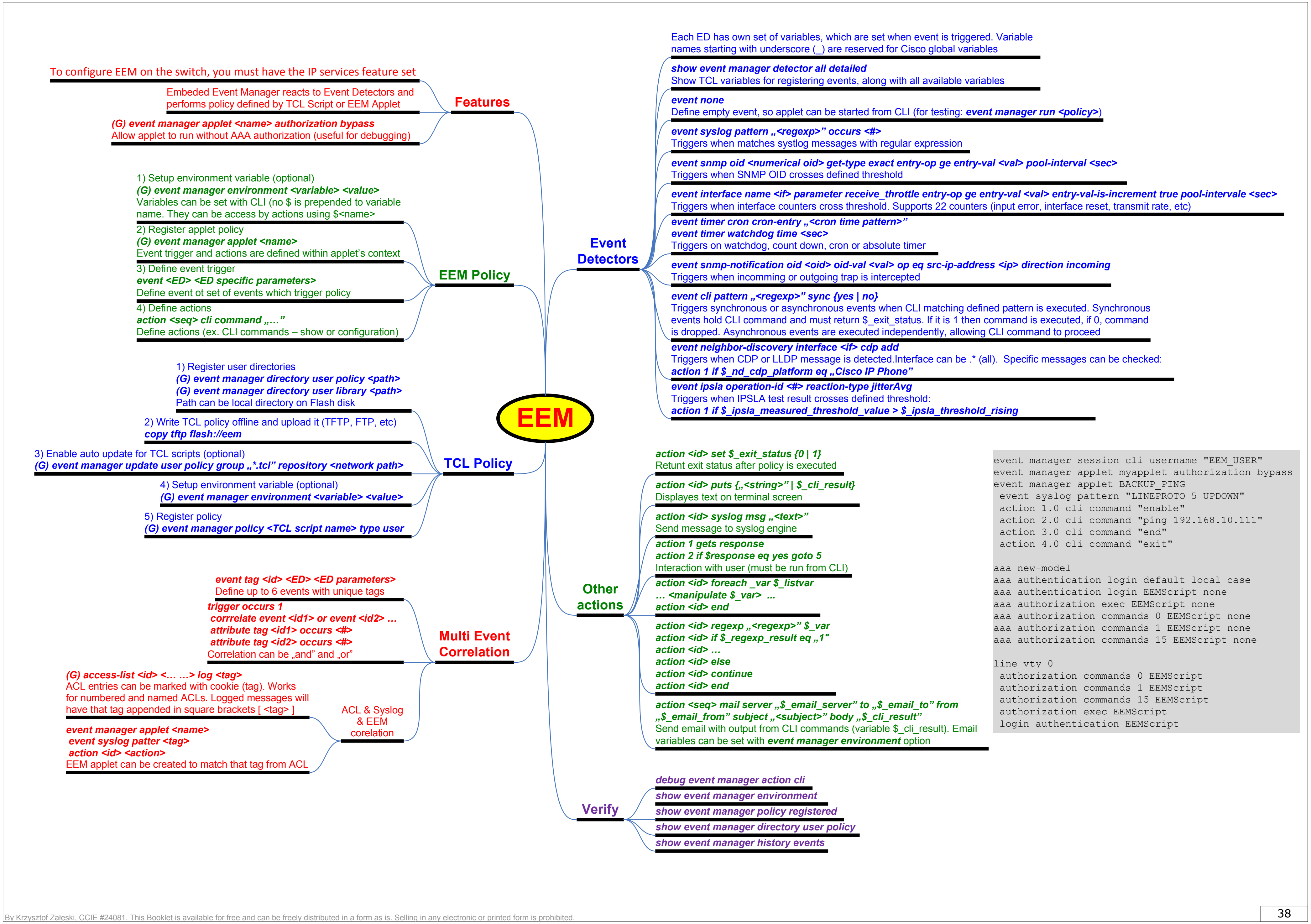
- Original version 1 is the default. Most common version is 5. Aggregation is possible in version 8 (11 schemas). All versions until 9 had fixed format, not compatible with each other. Flexible NetFlow is version 9
- Traditional NetFlow exports 7 key fields: Source IP, Destination IP, Source Port, Destination Port, L3 Protocol, TOS Byte (DSCP), Input interface. Provides packet and byte count
- ***show ip flow export***
- ***show ip cache [verbose] flow***
- ***ip flow-top-talkers***
  top <#>
  sort by {packets | bytes}
  match ...

Version 5

- ***(IF) ip flow {ingress | egress}***
  NetFlow will capture flows entering or leaving the router, but NOT to the router or from the router itself – only transiting traffic. Ingress flow is applied before rate limiting and decryption, egress flow is applied after rate limiting and encryption
- ***ip flow-export version 5 [origin-as | peer-as | bgp-nexthop]***
- ***ip flow-export destination <ip> <udp-port>***
- ***ip flow-cache entries <#>***
- ***ip flow-export source <if>***
- ***ip flow-cache timeout inactive <sec>***
  How long inactive flow will remain in cache before expiration (default 15 sec)
- ***ip flow-cache timeout active <sec>***
  How long active flow will remain in cache before expiration (default 30 min)
- ***(G) ip flow-capture {fragment-offset | icmp | ip-id | mac-addresses | packet-length | ttl | vlan-id | nbar}***
  Capture values from Layer 2 or additional Layer 3 fields
- ***(G) ip flow-export interface-names***
  Sends both: ifIndex and ifName in option data record

Version 9

- Version 9 defines exporting process with new aggregations. Flexible Netflow is an extension
- Template FlowSet and Data FlowSet. Template is composed of Type and Length, sent periodicaly
- ***(G) ip flow-export template options export-stats***
  Enable sending export statistics (total flows and packets exported) as options data
- ***(G) ip flow-export template [options] timeout-rate <#>***
  Templates and options sent every # of minutes
- ***(G) ip flow-export template [options] refresh-rate <#>***
  Templates and options sent every # of packets
- 1) Configure Template
  (G) flow record <name>
  - Two parameters: match and collect define what will be cought and included in the flow cache
- 2) Configure Exporter
  (G) flow exporter <name>
  - ***destination <ip>***
  - ***transport udp <port>***
  - ***export-protocol {netflow-v5 | netflow-v9}***
- 3) Configure Monitor
  (G) flow monitor <name>
  - ***exporter <name>***
  - ***record <name>***
  - ***cache entries <#>***
  - ***cache timeout {active | inactive | update} <sec>***
  - ***cache type {normal | immediate | permanent}***
    Normal – active and inactive timers. Immediate - all packets (real-time).
    Permanent – entire cache periodicaly exported; no monitoring when full
- 4) Configure interface
  (IF) ip flow monitor <name> {input | output}



EEM

Features

To configure EEM on the switch, you must have the IP services feature set

Embeded Event Manager reacts to Event Detectors and performs policy defined by TCL Script or EEM Applet

(G) event manager applet <name> authorization bypass
Allow applet to run without AAA authorization (useful for debugging)

EEM Policy

1) Setup environment variable (optional)
(G) event manager environment <variable> <value>
Variables can be set with CLI (no $ is prepended to variable name. They can be access by actions using $<name>

2) Register applet policy
(G) event manager applet <name>
Event trigger and actions are defined within applet's context

3) Define event trigger
event <ED> <ED specific parameters>
Define event ot set of events which trigger policy

4) Define actions
action <seq> cli command „…"
Define actions (ex. CLI commands – show or configuration)

TCL Policy

1) Register user directories
(G) event manager directory user policy <path>
(G) event manager directory user library <path>
Path can be local directory on Flash disk

2) Write TCL policy offline and upload it (TFTP, FTP, etc)
copy tftp flash://eem

3) Enable auto update for TCL scripts (optional)
(G) event manager update user policy group „*.tcl" repository <network path>

4) Setup environment variable (optional)
(G) event manager environment <variable> <value>

5) Register policy
(G) event manager policy <TCL script name> type user

Multi Event Correlation

event tag <id> <ED> <ED parameters>
Define up to 6 events with unique tags

trigger occurs 1
corrrelate event <id1> or event <id2> …
attribute tag <id1> occurs <#>
attribute tag <id2> occurs <#>
Correlation can be „and" and „or"

ACL & Syslog & EEM corelation

(G) access-list <id> <… …> log <tag>
ACL entries can be marked with cookie (tag). Works for numbered and named ACLs. Logged messages will have that tag appended in square brackets [<tag>]

event manager applet <name>
event syslog patter <tag>
action <id> <action>
EEM applet can be created to match that tag from ACL

Event Detectors

Each ED has own set of variables, which are set when event is triggered. Variable names starting with underscore (_) are reserved for Cisco global variables

show event manager detector all detailed
Show TCL variables for registering events, along with all available variables

event none
Define empty event, so applet can be started from CLI (for testing: ***event manager run <policy>***)

event syslog pattern „<regexp>" occurs <#>
Triggers when matches systlog messages with regular expression

event snmp oid <numerical oid> get-type exact entry-op ge entry-val <val> pool-interval <sec>
Triggers when SNMP OID crosses defined threshold

event interface name <if> parameter receive_throttle entry-op ge entry-val <val> entry-val-is-increment true pool-intervale <sec>
Triggers when interface counters cross threshold. Supports 22 counters (input error, interface reset, transmit rate, etc)

event timer cron cron-entry „<cron time pattern>"
event timer watchdog time <sec>
Triggers on watchdog, count down, cron or absolute timer

event snmp-notification oid <oid> oid-val <val> op eq src-ip-address <ip> direction incoming
Triggers when incomming or outgoing trap is intercepted

event cli pattern „<regexp>" sync {yes | no}
Triggers synchronous or asynchronous events when CLI matching defined pattern is executed. Synchronous events hold CLI command and must return $_exit_status. If it is 1 then command is executed, if 0, command is dropped. Asynchronous events are executed independently, allowing CLI command to proceed

event neighbor-discovery interface <if> cdp add
Triggers when CDP or LLDP message is detected.Interface can be .* (all). Specific messages can be checked:
action 1 if $_nd_cdp_platform eq „Cisco IP Phone"

event ipsla operation-id <#> reaction-type jitterAvg
Triggers when IPSLA test result crosses defined threshold:
action 1 if $_ipsla_measured_threshold_value > $_ipsla_threshold_rising

Other actions

action <id> set $_exit_status {0 | 1}
Retunt exit status after policy is executed

action <id> puts {„<string>" | $_cli_result}
Displayes text on terminal screen

action <id> syslog msg „<text>"
Send message to syslog engine

action 1 gets response
action 2 if $response eq yes goto 5
Interaction with user (must be run from CLI)

action <id> foreach _var $_listvar
… <manipulate $_var> …
action <id> end

action <id> regexp „<regexp>" $_var
action <id> if $_regexp_result eq „1"
action <id> …
action <id> else
action <id> continue
action <id> end

action <seq> mail server „$_email_server" to „$_email_to" from „$_email_from" subject „<subject>" body „$_cli_result"
Send email with output from CLI commands (variable $_cli_result). Email variables can be set with ***event manager environment*** option

Verify

debug event manager action cli
show event manager environment
show event manager policy registered
show event manager directory user policy
show event manager history events

```
event manager session cli username "EEM_USER"
event manager applet myapplet authorization bypass
event manager applet BACKUP_PING
 event syslog pattern "LINEPROTO-5-UPDOWN"
 action 1.0 cli command "enable"
 action 2.0 cli command "ping 192.168.10.111"
 action 3.0 cli command "end"
 action 4.0 cli command "exit"

aaa new-model
aaa authentication login default local-case
aaa authentication login EEMScript none
aaa authorization exec EEMScript none
aaa authorization commands 0 EEMScript none
aaa authorization commands 1 EEMScript none
aaa authorization commands 15 EEMScript none

line vty 0
 authorization commands 0 EEMScript
 authorization commands 1 EEMScript
 authorization commands 15 EEMScript
 authorization exec EEMScript
 login authentication EEMScript
```

Routing entry

Command (8)	Version (8)	All zeros (16)	
AFI (16)		Route TAG (16)	
Network (32			
Netmask (32)			
Next hop (32)			
Metric (32)			

20 B

Authentication entry

Command (8)	Version (8)	All zeros (16)
0xFFFF		Auth Type (16)
Authentication (128)		

RIPv2

Features

- ***(G) router rip***
  Only one, global session, no AS, name, etc
- Distance-vector (Bellman-Ford), standarized, some features still taken from RIPv1 (classful)
- Best path is a hop-count, loop prevention: split-horizon, poison-reverse, holddown-timers
- Updates sent to UDP/520. RIPv1 uses broadcast, RIPv2 uses 224.0.0.9. Unreliable (no ACK)
- Commands: Request (Type 1), Response (Type 2) – also known as Update, may be unicasted to the neighbor
- ***(IF) ip rip {send | receive} version 1 2***
  By default RIP sends only RIPv1 messages but listens to both RIPv1 and RIPv2. If version 2 is enabled globaly, only v2 updates are sent and received
- No neighbor relationship, no Hello
- ***(RIP) neighbor <ip>***
  Unicast updates to specified peer. Use 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.
- RIP is NFS-aware
- ***(IF) ip rip v2-broadcast***
  Behaves like RIPv1. Multicast messages are suppressed
- ***(RIP) passive interface {default <if>}***
  Disable sending updates, but still receives updates. To filter inbound updates distribute-list must be used
- BFD
  - ***(RIP) bfd all-interfaces***
  - ***(RIP) neighbor <ip> bfd***

Timers

- All timers start at the same time, they are not cumulative
- Update 30 sec – Random amount of time (Cisco IOS only) is subtracted from the update time. Up to 15 percent (4.5 seconds), so updates vary between 25.5 and 30 sec
- Invalid 180 sec – Route becomes invalid if no updates are heard within that time. Route is marked inaccessible (metric 16) and advertised as unreachable but router still uses it to forward packets
- Holddown 180 sec – If route's metric changes, do not accept sources of updates with worse metric (than original route's metric) until this timer expires. This timer is introduced by CISCO, it is not in RFC.
- Flush 240 sec – Route is removed from routing table it this timer expires. Starts at the sam time as Invalid timer, so route is flushed after 60 sec after invalid timer expires
- ***(RIP) timers basic <update> <invalid> <hold> <flush> <sleep ms>***
  Sleep – delays regular periodic update after receiveing a triggered update
- ***(RIP) flash-update threshold <sec>***
  If this amount of time or less is left before regular, full update, then triggered update is suppressed
- ***(RIP) output-delay <sec>***
  If multiple updates are to be sent, wait this time between packets
- ***(IF) ip rip advertise <sec>***
  Define update interval per interface
- ***(IF) ip rip initial-delay <sec>***
  Postpone sending initial MD5 packets (some devices require initial MD5 packes to have sequence 0, first packets could be dropped in the segment that is just starting). Default is no dalay
- ***(RIP) throttle***
  Requires ***output-delay*** command. Only one request for update per minute will be served

Updates

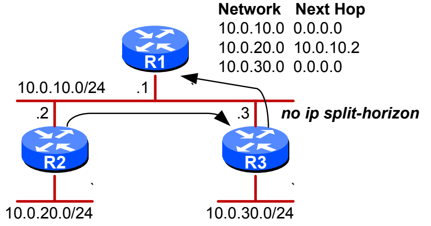
- ***(RIP) network x.x.x.x***
  Must be always in classful form (even in RIPv2), no netmask – IOS will convert automatically to classful. Secondary interface addresses can also besent in updates (must be covered with network statement). You can use ***network 0.0.0.0*** to include all interfaces
- Netmask does NOT have to be the same everywhere (network boundary or within a major network scope), to advertise v2 routes (netmask is carried in updates!)
- RIP advertises connected (covered by network statement) and other learned by RIP
- Each message can carry 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) – max UDP size (RFC)
- 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 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 (flush timer) expires (240 sec), then route will be removed from routing table
- ***(RIP) no validate-update-source***
  RIP and EIGRP are the only protocols that check source updates (if the same IP segment), 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.
- ***(RIP) input-queue <#>***
  RIP has internal queue for update packets. Default is 50 packets. In large RIP networks it may be required to increase it so there are no drops (no reliability in transport)
- Split horizon
  - ***(IF) no ip 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, neither autosummary nor summary-address from interface is advertised
  - By default ENABLED on multipoint sub-intf, but DISABLED on physical multipoint intf
  - If disabled, V1 and V2 can interoperate on the same interface
- Triggered
  - Suppresses periodic updates. Sends updates upon the change, and only the route that changed
  - Triggered are uni-directional (enabled on each side independently)
  - ***(IF) ip rip triggered***
    Available for WAN interfaces only. You MUST set **/30** subnet (/31 does not work) or you will see „invalid triggered header", and triggered updates are disabled. Usually used on on-demand circuits
  - If the router receives a request for a routing update full database is sent

Metric

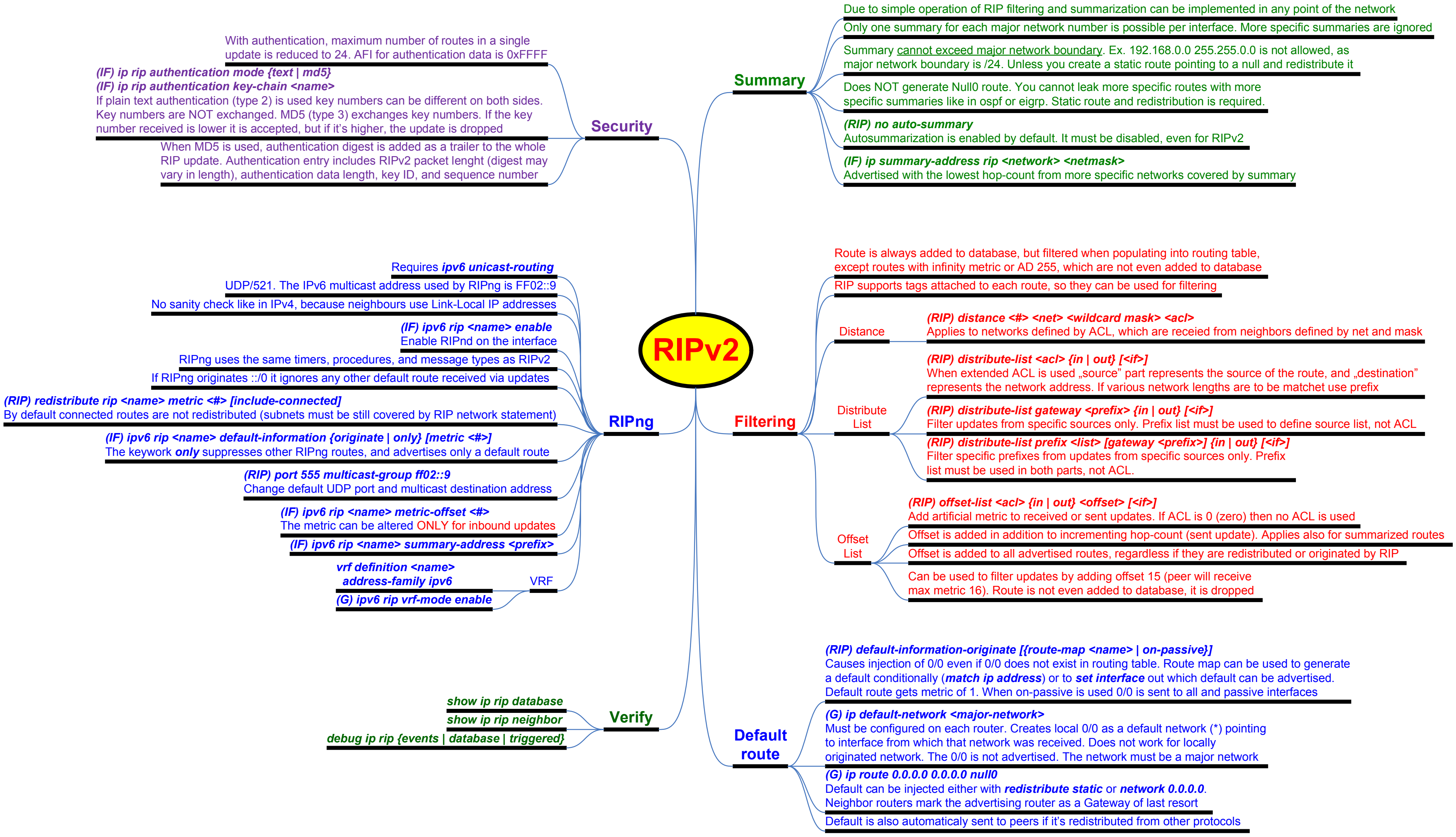
- Hop-count. Max 15 hops. Metric 16 means inaccessible and route is not placed into routing table
- Router adds 1 hop to each route sent to peers (localy connected routes have metric 0). This metric is installed in peer's routing table. Remote peer does not add a hop, unless offset-list is used
- ***(RIP) default-metric <#>***
  Define default seed metric for redistributed routes
- During redistribution from other protocols seed metric MUST be set manualy (***metric*** keyword or ***set metric*** inside ***route-map***). This manual metric is announced to peers as is. No additional hop is added when sending route to peers, unless offset-list is used

Next Hop

- Next-hop address of 0.0.0.0 specifies the originator of the update message
- Valid non-zero next-hop address specifies the next-hop router other than originator of the message (happens on shared subnet if a sending router has split-horizon disabled, and NH in update points to the other router which originated the update)



Network	Next Hop
10.0.10.0	0.0.0.0
10.0.20.0	10.0.10.2
10.0.30.0	0.0.0.0



RIPv2

Security

With authentication, maximum number of routes in a single update is reduced to 24. AFI for authentication data is 0xFFFF

(IF) ip rip authentication mode {text | md5}
(IF) ip rip authentication key-chain <name>
If plain text authentication (type 2) is used key numbers can be different on both sides. Key numbers are NOT exchanged. MD5 (type 3) exchanges key numbers. If the key number received is lower it is accepted, but if it's higher, the update is dropped

When MD5 is used, authentication digest is added as a trailer to the whole RIP update. Authentication entry includes RIPv2 packet lenght (digest may vary in length), authentication data length, key ID, and sequence number

RIPng

Requires ***ipv6 unicast-routing***

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

(IF) ipv6 rip <name> enable
Enable RIPnd on the interface

RIPng uses the same timers, procedures, and message types as RIPv2

If RIPng originates ::/0 it ignores any other default route received via updates

(RIP) redistribute rip <name> metric <#> [include-connected]
By default connected routes are not redistributed (subnets must be still covered by RIP network statement)

(IF) ipv6 rip <name> default-information {originate | only} [metric <#>]
The keywork ***only*** suppresses other RIPng routes, and advertises only a default route

(RIP) port 555 multicast-group ff02::9
Change default UDP port and multicast destination address

(IF) ipv6 rip <name> metric-offset <#>
The metric can be altered ONLY for inbound updates

(IF) ipv6 rip <name> summary-address <prefix>

VRF
- ***vrf definition <name>***
  address-family ipv6
- ***(G) ipv6 rip vrf-mode enable***

Verify

show ip rip database
show ip rip neighbor
debug ip rip {events | database | triggered}

Summary

Due to simple operation of RIP filtering and summarization can be implemented in any point of the network

Only one summary for each major network number is possible per interface. More specific summaries are ignored

Summary cannot exceed major network boundary. Ex. 192.168.0.0 255.255.0.0 is not allowed, as major network boundary is /24. Unless you create a static route pointing to a null and redistribute it

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.

(RIP) no auto-summary
Autosummarization is enabled by default. It must be disabled, even for RIPv2

(IF) ip summary-address rip <network> <netmask>
Advertised with the lowest hop-count from more specific networks covered by summary

Filtering

Route is always added to database, but filtered when populating into routing table, except routes with infinity metric or AD 255, which are not even added to database

RIP supports tags attached to each route, so they can be used for filtering

Distance
- ***(RIP) distance <#> <net> <wildcard mask> <acl>***
  Applies to networks defined by ACL, which are receied from neighbors defined by net and mask

Distribute List
- ***(RIP) distribute-list <acl> {in | out} [<if>]***
  When extended ACL is used „source" part represents the source of the route, and „destination" represents the network address. If various network lengths are to be matchet use prefix
- ***(RIP) distribute-list gateway <prefix> {in | out} [<if>]***
  Filter updates from specific sources only. Prefix list must be used to define source list, not ACL
- ***(RIP) 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
- ***(RIP) 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
- Offset is added in addition to incrementing hop-count (sent update). Applies also for summarized routes
- Offset is added to all advertised routes, regardless if they are redistributed or originated by RIP
- Can be used to filter updates by adding offset 15 (peer will receive max metric 16). Route is not even added to database, it is dropped

Default route

(RIP) default-information-originate [{route-map <name> | on-passive}]
Causes injection of 0/0 even if 0/0 does not exist in routing table. Route map can be used to generate a default conditionally (***match ip address***) or to ***set interface*** out which default can be advertised. Default route gets metric of 1. When on-passive is used 0/0 is sent to all and passive interfaces

(G) ip default-network <major-network>
Must be configured on each router. Creates local 0/0 as a default network (*) pointing to interface from which that network was received. Does not work for locally originated network. The 0/0 is not advertised. The network must be a major network

(G) 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 mark the advertising router as a Gateway of last resort

Default is also automaticaly sent to peers if it's redistributed from other protocols

EIGRP

Features

- Protocol 88 multicasted to 224.0.0.10. Updates are unicasted between neighbors
- EIGRP is a distance-vector-based protocol, also known as hybrid
- 3 tables: neighbor, topology, routing
- 8 packets based on TLV. Hello, Update, Ack, Query, Reply, Goodbye, SIA Query, SIA Reply
- Multi-VRF configuration (VRF must be created before adding to EIGRP)
- Functional components: Protocol-Dependent Modules, Reliable Transport Protocol (RTP), Neighbor Discovery/Recovery, Diffusing Update Algorithm (DUAL)
- AD internal 90, external 170, summary 5
- ***(IF) ip bandwidth-percent eigrp <process> <%>***
  EIGRP traffic uses max 50% of bandwidth for control traffic (not data). If BW was artificialy lowered, % can be more than 100%. When there are many neighbors on multipoint interfaces (mGRE/ DMVPN) shares available bendwidth between number of spokes – BW is divided between peers
- ***(EIGRP) network <net> <reverse mask>***
  If you specify a plain netmask, IOS detects that and changes it to correct reverse mask. All interfaces can be defined as 0.0.0.0 255.255.255.255 or 0.0.0.0 0.0.0.0
- Router ID
  - Router ID is derived from 1) manual ***router-id*** command, 2) highest IP on loppbacks, 3) highest IP on other interfaces
  - Originator's Router ID is included in external prefixes. If router receives external route with own ID, it discards it to prevent loops

Named mode

- ***router eigrp <name>***
  address-family ipv4 unicast autonomous-system <as>
  The name has only a local meaning, it is not advertised
- ***(EIGRP-AF) af-interface {default | <if>}***
  All interface-based options: passive, timers, etc
- ***(EIGRP) eigrp upgrade-cli <name>***
  Migrate classic mode to named mode (15.4S). No downtime, gracefull restart (NSF)
- Global parameters are configured either in SAFI mode or in ***topology base*** (default). Multitopology routing (MTR) allows different topologies based on some criteria (QoS). MTR is rarely used (***global-address-family*** in global config)
- If some AS number is used in named-mode, it cannot be used in classic mode (AS overlap) in the other process
- Compatible with classic-mode (mixed modes on different routers)

Neighbors

- Hello (keepalive) not acknowledged
- Must be in the same AS and K-values must match
- Source of Hello is primary IP on intf. If neighbor has IP from the same subnet as secondary, no neighborship forms
- ***(EIGRP) neighbor <ip> <intf>***
  Send hellos as unicast, and suppress sending and receiving any hellos via 224.0.0.10 on specified interface. Static configuration is required for all other peers on the same interface
- ***(EIGRP) passive-interface {default | <if>}***
  Stops sending and ignores hellos on specified interface
- *Peer restarted* – other router reset our neighborship
  Holding time expired – we didn't hear any EIGRP packet from the neighbor within a hold time
  Retry limit exceeeded – neighbor didn't ACK a pacteks after 16th retry
- ***show ip eigrp interface [detail]***
- ***show ip eigrp neighbor [detail]***
  - Queue count > 0 = convergence/communication problem

```
R4#show ip eigrp neighbors
EIGRP-IPv4 VR(core) Address-Family Neighbors for AS(10)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q   Seq
                                                   (sec)         (ms)        Cnt Num
1   10.0.45.5               Gi0/0                    14 00:00:09  324  2916  0   5
0   10.0.34.3               Gi2/0                    12 02:05:46  876  5000  0   15
```

Q cnt should never be >0 long time

Sequence, which neighbor appeared first

Seq seen from neighbor (header)

Header

Header

Version (8)	Opcode (8)	Checksum (16)
Flags (32)		
Sequence (32)		
Ack (32)		
AS (32)		

General EIGRP Parameters

Type=0x0001		Length (16)	
K1	K2	K3	K4
K5	K6 (wide metric)	Holdtime	

IP Internal Routes

Type=0x0102		Length (16)	
Next hop (32)			
Delay (32)			
Bandwidth (32)			
MTU (24)			Hop (8)
Reliab. (8)	Load (8)	Reserverd	
Prefix len (8)	Destination (0-padded)		

IP External Routes

Type=0x0103		Length (16)	
Next hop (32)			
Originating router ID (32)			
Originating AS (32)			
Tag (32)			
External protocol metric (32)			
Reserved (16)		Ext Proto ID	Flags (8)
Delay (32)			
Bandwidth (32)			
MTU (24)			Hop (8)
Reliab. (8)	Load (8)	Reserverd (16)	
Prefix len (8)	Destination (0-padded) (len vary)		

General TLV schema

Type high	Type low	Length (16)	
Value (variable length)			

- **Type high:** protocol (General, IPv4, IPv6, etc); **Type low:** TLV Op Code
- **Opcode**: 1: Update; 2: Reserved; 3: Query; 4: Reply; 5: Hello; 6: IPX-SAP; 10: SIA Query; 11: SIA Reply
- **TLV Types**: 0x0001: General EIGRP Parameters; 0x0002: Auth Type; 0x0003: Sequence; 0x0004: IOS and EIGRP code versions; 0x0005: Multicast Sequence; 0x0102: IP Internal Routes; 0x0103: IP External Routes
- **Header flags**. The right-most bit is Init, which indicates that the enclosed route entries are the first in a new neighbor relationship. The second bit is the Conditional Receive bit, used in Reliable Multicasting algorithm
- **Ext route flags**. The right-most bit indicates an external route. If the second bit is set, the route is a candidate default route

Timers

- ***(EIGRP-AF-IF) hello-interval <sec>***
  (EIGRP-AF-IF) hold-time <sec>
- ***(IF) ip hello-interval eigrp <process> <sec>***
  (IF) ip hold-time eigrp <process> <sec>
- Hello and Hold can be changed independently
- Holdtime is announced in Hello, but does not have to match. Router uses value announced by neighbor
  - NBMA: 60 sec / 180 sec
  - Other: 5sec / 15 sec
- Hold time is reset every time any EIGRP packet (not only Hello) is received
- ***(EIGRP) timers active-time {<sec> | disabled}***
  Default is 3 min. If no response to query is received within this time, the route is declared SIA

EIGRPv6

- Requires ***ipv6 unicast-routing***
- EIGRPv4 and EIGRPv6 are separate protocols
- Hellos are sent from link-local address to FF02::A (All EIGRP routers)
- ***(G) ipv6 router eigrp <as>***
- ***(IF) ipv6 eigrp <as>***
  EIGRPv6 is directly enabled on the interfaces. No ***network*** statement is used.
- ***(EIGRP) address-family ipv6 unicast autonomous-system <as>***
  Named mode uses own AF for IPv6. Can be configured in the same process as v4
- ***(EIGRP) eigrp router-id <ip>***
  Router ID is required, and it's still 32-bit address (used to identify the source of update, so IPv6 would limit the size of updates). If not defined, available IPv4 address is used (must be in the same VRF as IPv6)
- ***(EIGRP) no shutdown***
  When EIGRPv6 process is first enabled it is by default in shutdown mode
- All classic commands are exactly the same as in v4, just replace ***ip*** with ***ipv6***

EIGRP

Topology (DUAL)

- RD (reported distance) – distance reported by the peer
- Successor – peer that is currently being used as the next hop to the destination
- FD (feasible distance) – the best distance to remote network (successor route) installed in the routing table
- FS – feasible successor – not the best route, but still meets feasibility condition (RD < FD) – is closer to the destination than local router (no loop)
- Metrics for each route shown as: (Feasible distance / Reported distance)
- ***show ip eigrp topology all-links***
  Topology also contains non-feasible routes, but they are not used (AD < FD)
- Zero-successor route in topology means EIGRP tried to install route in RIB but there was another route already there with better AD. It can be also the case when there are two EIGRP processes. Only one can install route in RIB. Zero-successor routes are not propagated to peers
- If Successor disappears
  - **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). This is local computation
  - **2**. If no FS exists, router performs active query for the prefix. This is diffusing computation across domain.

RTP

- **Reliable Transport Protocol**
- Ordered delivery is provided by two sequence numbers. Each packet includes SN assigned by neighbor. It is incremented by one each time the router sends a new packet. Also, the sending router places in the packet the SN of last packet received from neighbor
- SRTT – how long does it take for a neighbor to respond to reliable packets. Derived from previous measurements of how long it took to get ACK. Each message, except Hello and ACK, has to be ACKed
- Multicast Flow Timer (***show ip eigrp interface***) – The time to wait for an ACK before switching from multicast to unicast. Calculated for each peer, from SRTT
- RTO – The time between the subsequent unicasts, when no ACK is received. Calculated for each peer, from SRTT
- If a packet is reliably multicasted and an ACK is not received from a neighbor, the packet will be retransmitted as a unicast to that neighbor. If an ACK is not received after 16th unicast retransmission, the neighbor will be declared dead
- Messages are multicasted with CR-bit set (Conditional Receive) with TLV listing peers which didn't send ACK (sequence TLV). Each retry backs-off 1.5 times the last interval. Min is 200ms, max i 5000 msec. When 5sec is reached it is repeated until 16th retry. Max retry period is 80 sec if starting with 5sec and 5sec consecutive dalays

```
R4#show ip eigrp topology 1.1.1.1 255.255.255.255
EIGRP-IPv4 VR(core) Topology Entry for AS(10)/ID(44.44.44.44) for 1.1.1.1/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 7864320, RIB is 61440
  Descriptor Blocks:
  10.0.34.3 (GigabitEthernet2/0), from 10.0.34.3, Send flag is 0x0
      Composite metric is (7864320/7208960), route is External
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 110000000 picoseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 33.33.33.33
      External data:
        AS number of route is 0
        External protocol is RIP, external metric is 1
        Administrator tag is 0 (0x00000000)
```

RIB cost after scaling

RD

External route = redistributed

Router which performed redistribution

Source protocol is passed to all peers

Metric of source protocol when redistributed

```
R4#show ip eigrp topology 3.3.3.3 255.255.255.255
EIGRP-IPv4 VR(core) Topology Entry for AS(10)/ID(44.44.44.44) for 3.3.3.3/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1392640, RIB is 10880
  Descriptor Blocks:
  10.0.34.3 (GigabitEthernet2/0), from 10.0.34.3, Send flag is 0x0
      Composite metric is (1392640/163840), route is Internal
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 11250000 picoseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 33.33.33.33
```

Internal route = network statement

Router which performed redistribution

```
R5#show ip route 1.1.1.1 255.255.255.255
Routing entry for 1.1.1.1/32
  Known via "eigrp 10", distance 170, metric 66560, type external
  Redistributing via eigrp 10
  Last update from 10.0.45.4 on GigabitEthernet0/0, 00:00:06 ago
  Routing Descriptor Blocks:
  * 10.0.45.4, from 10.0.45.4, 00:00:06 ago, via GigabitEthernet0/0
      Route metric is 66560, traffic share count is 1
      Total delay is 120 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

Route redistributed

Neighbor, from which update was received

Next Hop address (* - no DNS name)

Query

- All queries and replies must be ACKed (RTP)
- A query origin flag (O) is set to 1 by router originated query
- 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
- Query scoping is used to avoid SIA and to minimize convergence time
  - **Stub router** – is never asked for any route
  - **Route summarization** – peer with summarized route instantly replies negatively without doing own query
- **1)** Router multicasts **query** to all peers and sets a query origin flag (O) to 1 (router originated query)
- **2)** Each peer replies (unicast) if they have or not, a route to that prefix
- **3)** Router updates own tolopogy table only if all neighbors replied
- If peer doesn't have the route, it witholds reply and performs 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 for which he stared own query
- If router stays too long in active query the route becomes Stuck In Active (**SIA**)
  - After half of active time (default 90 sec) router which originated Query and didn't get Reply, sends SIA Query as a reminder
  - The neighbor replies (SIA Reply) if it still waits for his own queries
  - Query is sent 3 times, then route is marked SIA (neighbor is reset)
  - ***show ip eigrp topology active***

EIGRP

Redistribution and filtering

- In named mode redistribution is done in topology (base)
- Seed metric must be set for routes distributed into EIGRP
- ***(EIGRP) redistribute <protocol> metric <bw> <delay> <reliability> <load> <mtu>***
- ***(EIGRP) default-metric <bw> <delay> <reliability> <load> <mtu>***
  Define default metric for all networks redistributed from other routing protocols (only)
- Metric is derived automaticaly only for routes redistributed from static, connected or other EIGRP processes. Static metric is derived from next-hop interface (must be covered with ***network***)
- When static route points to local interface (also null0), it is a pseudo-connected. It can be then picked up by EIGRP with network statement. It is seen as internal route. But it is NOT redistributed with ***redistribute connected***. However, if stub is configured, eigrp requires ***eigrp stub connected static***
- ***(EIGRP) distribute-list <acl> {in [<if>] | out [<if> | <protocol>]}***
  (EIGRP) distribute-list prefix <name> {in [<if>] | out [<if> | <protocol>]}
  (EIGRP) distribute-list route-map <name> {in [<if>] | out [<if> | <protocol>]}
  Protocol: <u>to</u> which redistribution is performed
- ***(EIGRP) distribute-list gateway <prefix-list> {in [<if>] | out [<if> | <protocol>]}***
  Filer routes based on peer's (gateway) IP. Prefix list defines gateway IP, not networks received
- Extended ACL in IGPs define source of update in the source part of ACL and networks in the destination part of ACL
- ***(IF) no ip next-hop-self eigrp <as>***
  By default, when routes are redistributed into EIGRP, and they are passed to EIGRP peers, router sets own outgoing interface's IP address as next-hop. If disabled, NH is coppied from other routing protocols (OSPF, RIP, but NOT BGP)
- ***(RM) match ip route-source <acl> <acl> ...***
- ***(RM) match source-protocol <proto> [<as>]***
  Valid protocols: bgp, connected, eigrp, isis, ospf, rip, and static

Route Tag

- ***(G) route-tag notation dotted-decimal***
  Change TAG notation from integer to dotted-decimal
- ***(RM) match tag <#>*** ***(RM) set tag <#>***
- ***(G) route-tag list <name> {deny | permit} <tag> <wildcard mask>***
  Tak must be in dotted decimal format. Supported in named mode
- ***(RM) match tag list <name>***
  Only matching is supoprted for TAG list
- ***(EIGRP-AF) eigrp default-route-tag <tag>***
  Set tag for all <u>internal</u> routes
- ***show ip route tag***

Max Prefix

- Supported only for IPv4 per VRF address family
- ***(EIGRP-AF) neighbor maximum-prefix <#> [<threshold>] [[dampened] [reset-time <min>] [restart <min>] [restart-count <#>] | warning-only]***
  When defined in global mode and limit is exceeded, all sessions are torn down
- ***(EIGRP-AF) redistribute maximum-prefix <#> ...***
  In named mode configured in topology. Applies to redistributed routes only
- ***(EIGRP-AF) maximum-prefix <#> ...***
  In named mode configured in topology. Applies to routes from all sources
- ***(EIGRP-AF) neighbor <ip> maximum-prefix <#> [<threshold>] [warning-only]***
- Restart timer: how long the router will wait to form adjacency or accept redistributed routes after max limit has been exceeded. Default is 5 min
- Restart counter: number of times a peering session can be automatically reestablished or redistributed routes can be automatically relearned due to max limit exceeded. Then, you have to clear routes (*) or sessions manually. Default is 3
- Reset timer: reset the restart counter to 0 after reset-time period has expired. Controls long-term accumulated penalties. Default is 15 min
- Dampening: apply exponential penalty to the restart-time each time max limit is exceeded. Half-life for the decay is 150% of the restart-time. Suppress unstable peers. Disabled by default
- When ***warning-only*** is used only syslog messages are generated
- ***show ip eigrp accounting***

Distance

- ***(EIGRP) distance eigrp <internal> <external>***
  Distance set for all internal and external prefixes
- ***(EIGRP) distance <distance> <source IP> <source mask> [<acl>]***
  Set for prefixes originated by a source **ONLY** for internal routes, external are not matched at all

Metric

- K metrics must match to form adjacency
- Values do not have to be 1, they can be any number (plain math calculation)
- Internal paths are prefered over external paths regardless of metric
- If network has mixed EIGRP versions suboptimal paths may exist (named EIGRP activates wide metric for specified AS only)
- Router uses own interface bandwidth if it's lower than advertised by peer (lowest path BW is used)
- MTU is NOT a part of calculation. It is in the formula, but different MTUs do not influence ECMP on local router
- ***(EIGRP) metric weights <tos> <k1> <k2> <k3> <k4> <k5> <k6>***
  Defaul TOS=0 (always); **K1 (BW)=1**; K2 (Load)=0; **K3 (DLY)=1**; K4 (Rerliability)=0; K5 (MTU)=0; K6(Ext)=0 (extra attribute, currently not used, may be used in the future)
- ***(IF) delay <10ths of usec>***
  Delay set to 1 means 10 microseconds = 10.000.000 ps for calculations. Delay is a cumulative
- Default interface delays for interfaces below 1G cannot be set manually using wide metric (value 1 means 10.000.000 ps)
- Loopback: 1.250.000 ps; Gigabit: 10.000.000 ps (delay 1 on interface); Fast: 100.000.000 ps
- Reliability is a number between 1 and 255 that reflects the total outgoing error rates of the interfaces along the route, calculated on a five-minute average. 255 indicates a 100 percent reliable link
- ***(EIGRP) offset-list <acl> {in | out} <offset> [<if>]***
  Offset list adds specified value to a **delay** <u>before</u> local calculation is performed.
  Offset with interface takes precedence over generic offset (only one is added)
- ***(EIGRP) metric maximum-hop 1***
  You can filter prefixes to be announced only to nearest peer. Default hop-count is 100. Connected routes are announced with hop-count 0

Route-map

- ***(RM) set metric <bw in K> <delay> <reliability> <load> <mtu>***
- ***(RM) match metric [external] <#> <#> ...***
  There can be many metrics defined in one line (they are ORed). By default only internal routes are checked unless ***external*** is added
- ***(RM) match metric 400 +- 100***
  Matches metric from 300 to 500

- ***show ip eigrp topology <prefix>***

Classic metric

- Bandwidth: lowest BW inversed, multiplied by 10^7*256
  For 100.000 kbps (100M): 1/100.000 (inverse) * 10.000.000*256 = 25.600
- Delay: in 10ths of microsecond multiplied by 256
- Since scaling is 10^7, if we pass 1G, all calculations are the same. 10G link is treated the same as 40G link in ECMP. The same with delay, all links > 1G have 10us

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

Wide metric (named mode)

- Bandwidth (throughput): lowest BW inversed, multiplied by 10^7*65536
  For 10.000.000 kbps (10G): 1/10.000.000 (inverse) * 10.000.000*65536 = 65536
- Delay (latency)
  - Below 1G: (Delay*65536)/10
  - Above and equal 1G: picoseconds multiplied by 65538, and divided by 10^6
- ***(EIGRP AF) metric rib-scale <1-255>***
  Introduced <u>local</u> RIB scale. Default is 128. Wide composite metric sometimes does not fit in RIB (32bit). Metric in topology table is different than in routing table after scaling

$$\text{Metric} = (K1*BW + \frac{K2 * BW}{256 - Load} + K3*Delay + (K6 * Ext)) * \frac{K5}{Reliability + K4}$$

EIGRP

Next Hop

(IF) no ip next-hop-self eigrp <as> - only for classic process, won't work for AS defined in named mode
(EIGRP-AF-IF) no next-hop-self

If NH is set to 0.0.0.0, then use address of the router from which update was received (hub), otherwise, use 3rd party NH (other spoke). By default EIGRP changes NH to 0.0.0.0 when sending updates to other routers

Works only on shared media (Ethernet, DMVPN), along with ***no split-horizon***

(IF) no ip split-horizon eigrp <as>
(EIGRP-AF-IF) no split-horizon
Enabled by default (except on physical FR). Changing the mode resets nejghbors on that intf. Since EIGRP uses Feasibility Condition as loop prevention, split-horizon is just a way of limiting unnecessary updates

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 (can be network statement or redistribute static)

(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. Default AD for summary is 5

(G) ip default-network <classful network>
If defined, it will be set as candidate default in EIGRP. This network must be in topology table

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

(EIGRP) default-information {allowed {in | out} | in | out} [<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

Tagging default route is not supported

Load balancing

(EIGRP) maximum-paths <1..32>
By default EIGRP will load balance across 4 equal paths. The newest IOS codes support 32 paralel paths

(EIGRP) traffic-share min – send traffic over lowest-cost path only

(EIGRP) traffic-share min across-interfaces
If more paths exist than allowed choose the ones over different physical interfaces

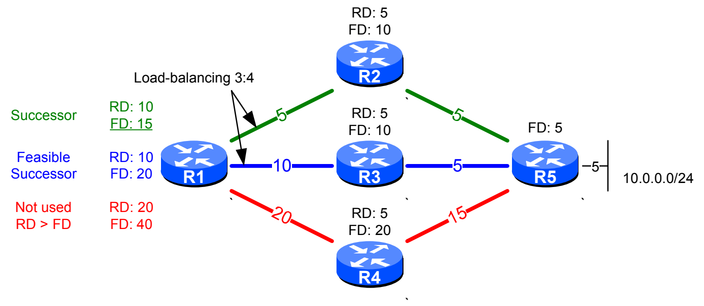
(EIGRP) traffic-share balanced – less packets to lower-bandwidth paths (default)

(EIGRP) variance <multiplier>
Multiplier is multiplied by FD (to get the variance divide the worst route by the FD and roun to upper integer). Any metric which is lower than this value and meets FC is also considered as valid load-balanced path. Traffic is shared in proportion to metrics (CEF assignes appropriate buckets)

(EIGRP) variance 2
Variance 2 in the below example means that any route with FD < 30 (2 * 15) will be used to load-balance traffic

Alternate path must meet Feasibility Condition

In named mode, parameters configured in topology (base)



Summary

(EIGRP) no auto-summary
Autosummarization is enabled by default up to 12.4T. It is off since 15.0. Autosummarization is done only on major network boundary, in regards to localy attached interface IP addresses, not prefixes received via updates (which could not be summarized if autosummary is not consistent through AS)

(IF) ip summary-address eigrp <as> <network> <mask> [<distance>] [leak-map <name>]
Default AD for summary is 5. Route is pointed to Null0. Metric is derived from lowest metric of component routes. 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. Summarization of all prefixes into 0.0.0.0/0 is possible

If component route flaps, summary also flaps and summary's metric must be recalculated. Router constantly checks topology table if best component route didn't change. It is recommended to use loopback interface to force the metric to remain constant (use delay to assign low metric)

(EIGRP) summary-metric <net> <mask> [<bw> <delay> <reliability> <load> <mtu>] [distance <ad>]
Define static metric for summary so CPU is not consumed when constantly checking topology table

Route leaking

Use ***leak-map*** to advertise suppressed routes. Not available on subinterfaces – use PPP and VirtualTemplate physical interface instead

More specific prefix can be also leaked with more specific summary route. Both leak-map and more specific summary can co-exst together.

(RM) match ip address <acl>
(G) access-list <acl> permit <net> <mask>
Routes permited by ACL will be leaked. If route-map does not exist, there is no leakinkg, but if ACL does not exist, summary and all component routes are sent

Stub router

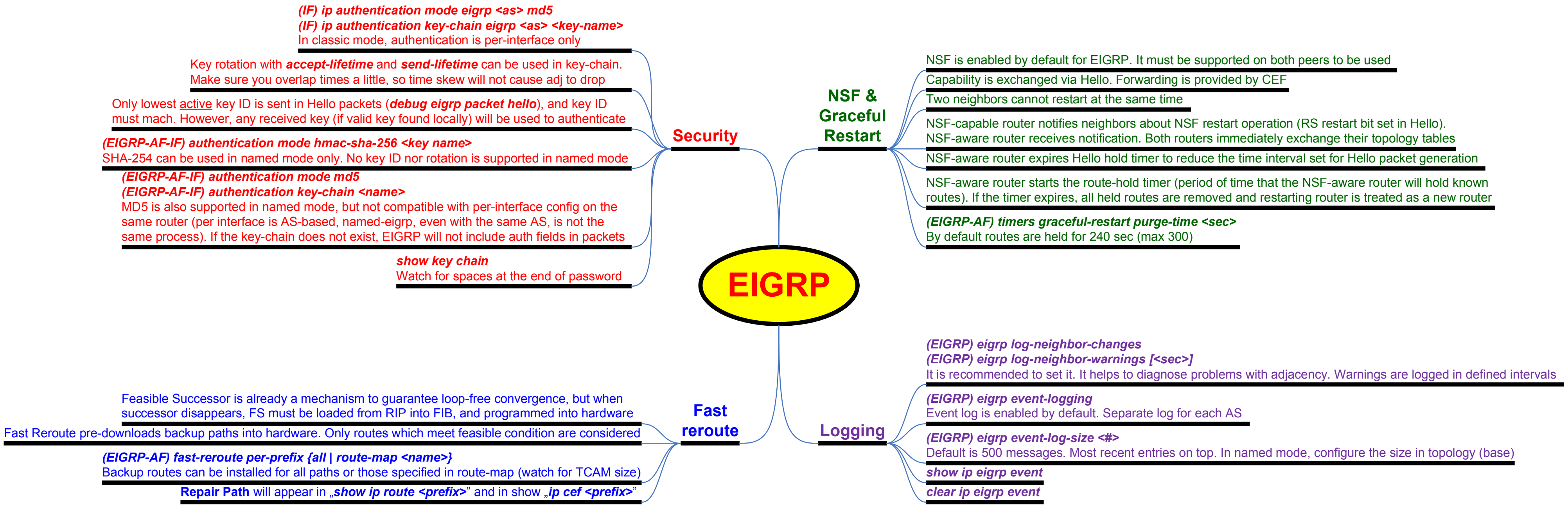
(EIGRP) eigrp stub {connected summary static redistributed receive-only} [leak <route-map>]
Stub by default announces connected and summary. Connected means covered by network statement or redistributed as connected. Redistributed routes cover only those not covered by network statement.

Routers do not query stub routers at all. Stub is announced in Hello

Stub routers cannot be used as transit. Prefixes learned via EIGRP are not propagated to other routers

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

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



EIGRP

Security

(IF) ip authentication mode eigrp <as> md5
(IF) ip authentication key-chain eigrp <as> <key-name>
In classic mode, authentication is per-interface only

Key rotation with ***accept-lifetime*** and ***send-lifetime*** can be used in key-chain. Make sure you overlap times a little, so time skew will not cause adj to drop

Only lowest active key ID is sent in Hello packets (***debug eigrp packet hello***), and key ID must mach. However, any received key (if valid key found locally) will be used to authenticate

(EIGRP-AF-IF) authentication mode hmac-sha-256 <key name>
SHA-254 can be used in named mode only. No key ID nor rotation is supported in named mode

(EIGRP-AF-IF) authentication mode md5
(EIGRP-AF-IF) authentication key-chain <name>
MD5 is also supported in named mode, but not compatible with per-interface config on the same router (per interface is AS-based, named-eigrp, even with the same AS, is not the same process). If the key-chain does not exist, EIGRP will not include auth fields in packets

show key chain
Watch for spaces at the end of password

NSF & Graceful Restart

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

Two neighbors cannot restart at the same time

NSF-capable router notifies neighbors about NSF restart operation (RS restart bit set in Hello). NSF-aware router receives notification. Both routers immediately exchange their topology tables

NSF-aware router expires Hello hold timer to reduce the time interval set for Hello packet generation

NSF-aware router starts the route-hold timer (period of time that the NSF-aware router will hold known routes). If the timer expires, all held routes are removed and restarting router is treated as a new router

(EIGRP-AF) timers graceful-restart purge-time <sec>
By default routes are held for 240 sec (max 300)

Fast reroute

Feasible Successor is already a mechanism to guarantee loop-free convergence, but when successor disappears, FS must be loaded from RIP into FIB, and programmed into hardware

Fast Reroute pre-downloads backup paths into hardware. Only routes which meet feasible condition are considered

(EIGRP-AF) fast-reroute per-prefix {all | route-map <name>}
Backup routes can be installed for all paths or those specified in route-map (watch for TCAM size)

Repair Path will appear in „***show ip route <prefix>***" and in show „***ip cef <prefix>***"

Logging

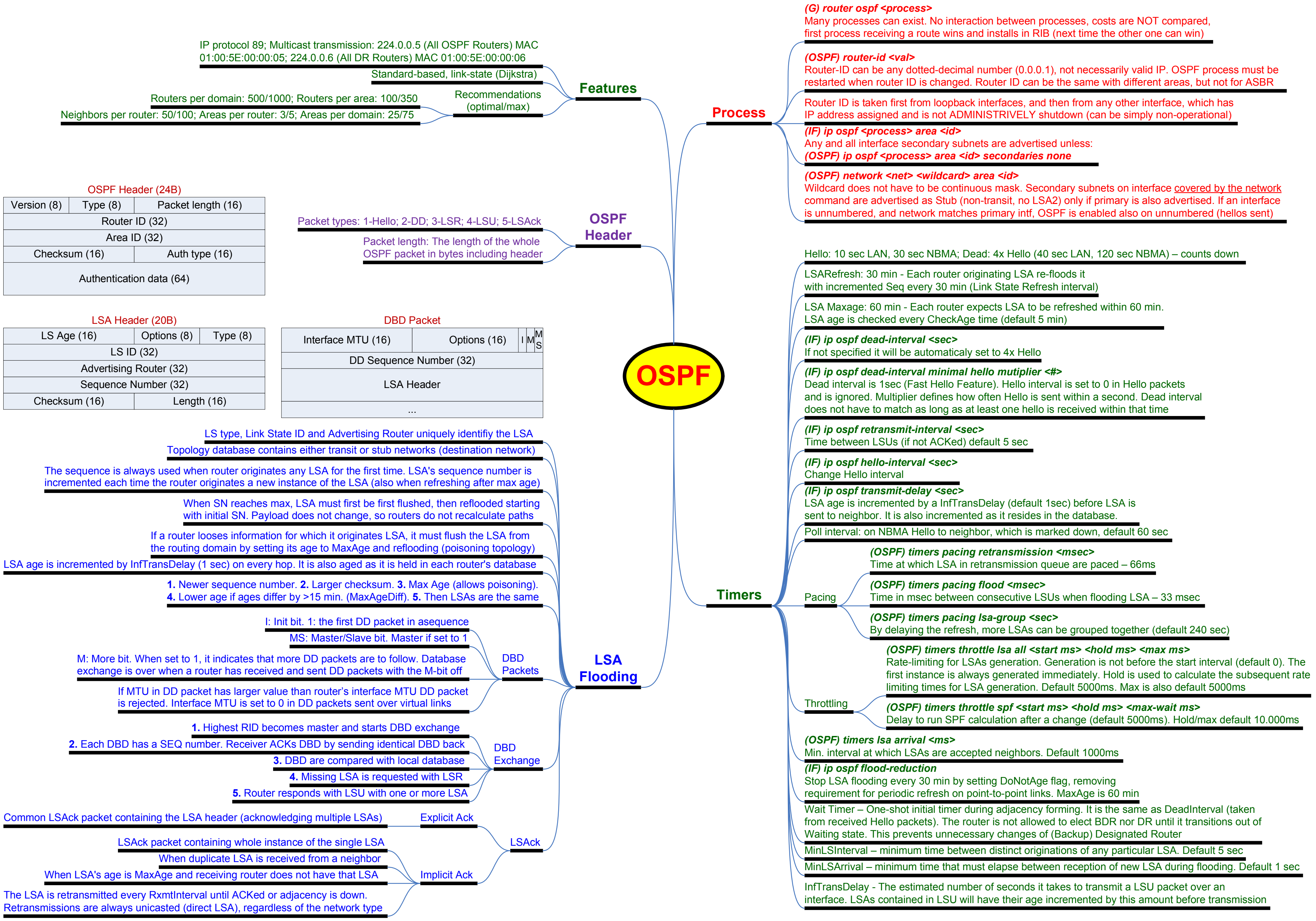
(EIGRP) eigrp log-neighbor-changes
(EIGRP) eigrp log-neighbor-warnings [<sec>]
It is recommended to set it. It helps to diagnose problems with adjacency. Warnings are logged in defined intervals

(EIGRP) eigrp event-logging
Event log is enabled by default. Separate log for each AS

(EIGRP) eigrp event-log-size <#>
Default is 500 messages. Most recent entries on top. In named mode, configure the size in topology (base)

show ip eigrp event

clear ip eigrp event



OSPF

Features

IP protocol 89; Multicast transmission: 224.0.0.5 (All OSPF Routers) MAC 01:00:5E:00:00:05; 224.0.0.6 (All DR Routers) MAC 01:00:5E:00:00:06

Standard-based, link-state (Dijkstra)

Recommendations (optimal/max)

- Routers per domain: 500/1000; Routers per area: 100/350
- Neighbors per router: 50/100; Areas per router: 3/5; Areas per domain: 25/75

OSPF Header

Packet types: 1-Hello; 2-DD; 3-LSR; 4-LSU; 5-LSAck

Packet length: The length of the whole OSPF packet in bytes including header

OSPF Header (24B)

Version (8)	Type (8)	Packet length (16)
Router ID (32)		
Area ID (32)		
Checksum (16)		Auth type (16)
Authentication data (64)		

LSA Header (20B)

LS Age (16)	Options (8)	Type (8)
LS ID (32)		
Advertising Router (32)		
Sequence Number (32)		
Checksum (16)	Length (16)	

DBD Packet

Interface MTU (16)	Options (16)	I	M	MS
DD Sequence Number (32)				
LSA Header				
...				

LSA Flooding

LS type, Link State ID and Advertising Router uniquely identifiy the LSA

Topology database contains either transit or stub networks (destination network)

The sequence is always used when router originates any LSA for the first time. LSA's sequence number is incremented each time the router originates a new instance of the LSA (also when refreshing after max age)

When SN reaches max, LSA must first be first flushed, then reflooded starting with initial SN. Payload does not change, so routers do not recalculate paths

If a router looses information for which it originates LSA, it must flush the LSA from the routing domain by setting its age to MaxAge and reflooding (poisoning topology)

LSA age is incremented by InfTransDelay (1 sec) on every hop. It is also aged as it is held in each router's database

1. Newer sequence number. **2.** Larger checksum. **3.** Max Age (allows poisoning). **4.** Lower age if ages differ by >15 min. (MaxAgeDiff). **5.** Then LSAs are the same

DBD Packets

- I: Init bit. 1: the first DD packet in asequence
- MS: Master/Slave bit. Master if set to 1
- M: More bit. When set to 1, it indicates that more DD packets are to follow. Database exchange is over when a router has received and sent DD packets with the M-bit off
- If MTU in DD packet has larger value than router's interface MTU DD packet is rejected. Interface MTU is set to 0 in DD packets sent over virtual links

DBD Exchange

- **1.** Highest RID becomes master and starts DBD exchange
- **2.** Each DBD has a SEQ number. Receiver ACKs DBD by sending identical DBD back
- **3.** DBD are compared with local database
- **4.** Missing LSA is requested with LSR
- **5.** Router responds with LSU with one or more LSA

LSAck

- Explicit Ack
  - Common LSAck packet containing the LSA header (acknowledging multiple LSAs)
- Implicit Ack
  - LSAck packet containing whole instance of the single LSA
  - When duplicate LSA is received from a neighbor
  - When LSA's age is MaxAge and receiving router does not have that LSA
  - The LSA is retransmitted every RxmtInterval until ACKed or adjacency is down. Retransmissions are always unicasted (direct LSA), regardless of the network type

Process

(G) router ospf <process>
Many processes can exist. No interaction between processes, costs are NOT compared, first process receiving a route wins and installs in RIB (next time the other one can win)

(OSPF) router-id <val>
Router-ID can be any dotted-decimal number (0.0.0.1), not necessarily valid IP. OSPF process must be restarted when router ID is changed. Router ID can be the same with different areas, but not for ASBR

Router ID is taken first from loopback interfaces, and then from any other interface, which has IP address assigned and is not ADMINISTRIVELY shutdown (can be simply non-operational)

(IF) ip ospf <process> area <id>
Any and all interface secondary subnets are advertised unless:
(OSPF) ip ospf <process> area <id> secondaries none

(OSPF) network <net> <wildcard> area <id>
Wildcard does not have to be continuous mask. Secondary subnets on interface covered by the network 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 sent)

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 it with incremented Seq every 30 min (Link State Refresh interval)

LSA Maxage: 60 min - Each router expects LSA to be refreshed within 60 min. LSA age is checked every CheckAge time (default 5 min)

(IF) ip ospf dead-interval <sec>
If not specified it will be automaticaly set to 4x Hello

(IF) ip ospf dead-interval minimal hello mutiplier <#>
Dead interval is 1sec (Fast Hello Feature). Hello interval is set to 0 in Hello packets and is ignored. Multiplier defines how often Hello is sent within a second. Dead interval does not have to match as long as at least one hello is received within that time

(IF) ip ospf retransmit-interval <sec>
Time between LSUs (if not ACKed) default 5 sec

(IF) ip ospf hello-interval <sec>
Change Hello interval

(IF) ip ospf transmit-delay <sec>
LSA age is incremented by a InfTransDelay (default 1sec) before LSA is sent to neighbor. It is also incremented as it resides in the database.

Poll interval: on NBMA Hello to neighbor, which is marked down, default 60 sec

Pacing

- ***(OSPF) timers pacing retransmission <msec>***
  Time at which LSA in retransmission queue are paced – 66ms
- ***(OSPF) timers pacing flood <msec>***
  Time in msec between consecutive LSUs when flooding LSA – 33 msec
- ***(OSPF) timers pacing lsa-group <sec>***
  By delaying the refresh, more LSAs can be grouped together (default 240 sec)

Throttling

- ***(OSPF) timers throttle lsa all <start ms> <hold ms> <max ms>***
  Rate-limiting for LSAs generation. Generation is not before the start interval (default 0). The first instance is always generated immediately. Hold is used to calculate the subsequent rate limiting times for LSA generation. Default 5000ms. Max is also default 5000ms
- ***(OSPF) timers throttle spf <start ms> <hold ms> <max-wait ms>***
  Delay to run SPF calculation after a change (default 5000ms). Hold/max default 10.000ms

(OSPF) timers lsa arrival <ms>
Min. interval at which LSAs are accepted neighbors. Default 1000ms

(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. MaxAge is 60 min

Wait Timer – One-shot initial timer during adjacency forming. It is the same as DeadInterval (taken from received Hello packets). The router is not allowed to elect BDR nor DR until it transitions out of Waiting state. This prevents unnecessary changes of (Backup) Designated Router

MinLSInterval – minimum time between distinct originations of any particular LSA. Default 5 sec

MinLSArrival – minimum time that must elapse between reception of new LSA during flooding. Default 1 sec

InfTransDelay - The estimated number of seconds it takes to transmit a LSU packet over an interface. LSAs contained in LSU will have their age incremented by this amount before transmission

OSPF

Neighbor

Hello

Hello

Network Mask (32)		
Hello interval (16)	Options (8)	Priority (8)
Dead interval (32)		
DR (32)		
BDR (32)		
Neighbor router ID		
...		

Options:

-	-	DC	EA	NP	MC	E	-

E: LSA5 is upported on thet interface
MC: Multicast send using RFC 1584
N: Type-7 LSA supported in area
P: NSSA ABR should translate 7>5
EA: External LSAs are supported in area
DC: Demand circuits capability

Sourced from interface primary subnet

Sent to 224.0.0.5 MAC:0100.5E00.0005

Adjacency

Adjacency is possible on unnumbered interfaces with different subnets but only if those interface are in the same area. Primary interface must be covered by network statement not an ***ip ospf*** interface command which is not inherited by unnumbered interface

If network statements overlap, most specific are used first to select area for an interface. Network statements are sorted automaticaly by IOS

To form an adjacency parameters must match: Authentication, Area number and type, Timers, Netmask, Stub flags, MTU

On p2p networks and virtual links, the Network Mask in the received Hello Packet is ignored

States

Attempt - applies only to manually configured neighbors on NBMA networks. A router sends packets to a neighbor at Poll Interval instead of Hello Interval

Init - Hello packet has been seen from the neighbor, but own Router ID is not yet present

2-Way - router has seen its own Router ID in the Neighbor field of the neighbor's Hello packets. DROTHER routers in broadcast networks remain in this state, which is valid (no full adjacency, only neighborship)

ExStart - routers establish a master/slave relationship and determine the initial DD sequence number. Highest Router ID becomes the master. DD header contains MTU. In MTUs are different, the one with lower MTU gets stuck in ExStart. MTU can be changed with ***ip mtu <mtu>***, but ***ip ospf mtu-ignore*** is recommended

Exchange - routers send DD packets with LSA headers to compare own databases

Loading - routers send LSR and LSU packets (full LSA exchange)

Full – routers reach full adjacency, databases are identical (per area)

```
R3#sh ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
5.5.5.5           0   FULL/  - p2p link 00:00:31  10.0.35.5       GigabitEthernet1/0
2.2.2.2           0   FULL/  -        00:00:33    10.0.23.2       GigabitEthernet2/0
1.1.1.1           1   FULL/BDR        00:00:37    10.0.123.1      GigabitEthernet0/0
2.2.2.2           1   FULL/DR         00:00:38    10.0.123.2      GigabitEthernet0/0
6.6.6.6           1   EXCHANGE/DR     00:00:35    10.0.46.6       GigabitEthernet2/0
 Router ID     Possible MTU issue  Neighbor's role   IP address on a segment
```

```
R3#sh ip ospf interface brief
Interface    PID   Area         IP Address/Mask    Cost  State Nbrs F/C
Lo0          1     0            3.3.3.3/24         1     P2P   0/0  F: fully adjacent
Gi1/0        1     0            10.0.35.3/24       1     P2P   1/1  C: in 2-way state
Gi2/0        1     0            10.0.23.3/24       1     P2P   1/1
Gi0/0        1     1            10.0.123.3/24      1     DROTH 2/2
          Process ID                                   Local cost
```

Authentication

Type0 – none (default), type1 – plain text, type2 – md5/sha (cryptographic authentication)

Every packet is authenticated (but nor encrypted)

All routers in area must be enabled for authentication (if per-area authentication is used), but not all links must have password set (only link which need to be protected). All routers within an area are not required to have authentication enabled if per-interface authentication is used

(IF) ip ospf authentication null
Type 0. Used to disable authentication on one interface

(IF) ip ospf authentication
(OSPF) area <id> authentication
Enable plain text authentication per interface or per area

(IF) ip ospf authentication-key <value>
Plain text password is always configured per interface

If plain text is used, whole authentication data is used to carry the password (max 8 charactes)

If MD5 is used, authentication data has different meaning (below)

Cryptographic sequence number is an unsigned non-decreasing number (increasing by 1, starting from 0), used to guard against replay attacks

All zeros (16)	Key ID (8)	Len (8)
Cryptographic Sequence Number (32)		

If multiple keys are configured on interface, multiple consecutive hellos are sent with all md5 digests until other side sends the matching key. If other side matches at least one key, adjacency stays up. If both sides are configured with new key, old ones are suppressed

The message digest itself is appended to the OSPF packet, but not considered as part of the OSPF packet (not included in header's length), but included in IP header length field

(IF) ip ospf authentication message-digest
(OSPF) area <id> authentication message-digest
Enable MD5 authentication per interface or per area

(IF) ip ospf message-digest-key <key#> md5 <key value>
Multiple keys can be configured to support key rotation or multiple peers on one interface

(IF) ip ospf authentication key-chain <key>
Auth type and password defined with one command. HMAC-SHA can be used only per interface. Not supported per-area

(KEY) cryptographic-algorithm hmac-sha-256

GTSM

Generic TTL Security Mechanism. By default TTL is set to 255, and verified by the peer (one hop allowed)

GTSM uses reverse logic. Routing protocols send packets with an IP TTL=255, not 1. Every router in the path decrements TTL by 1, so the number of hops can be easily calculated

(IF) ip ospf ttl-security [disable | hops <#>]
Accept OSPF packets with TTL = 256 – hop count. Available only for IPv4 (OSPFv2)

(OSPF) ttl-security all-interfaces [hops <#>]

OSPF

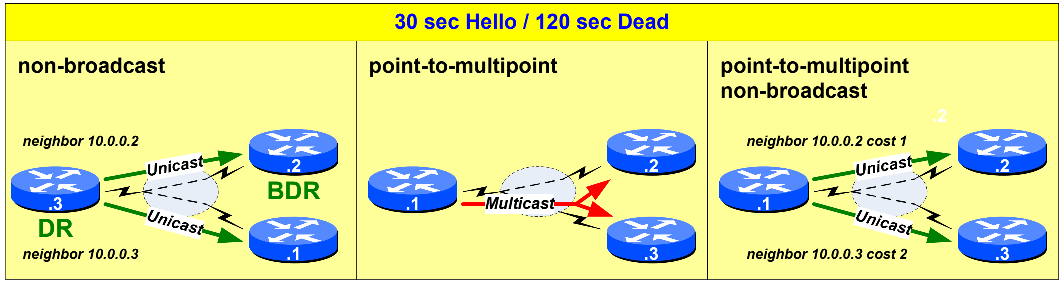
Network types

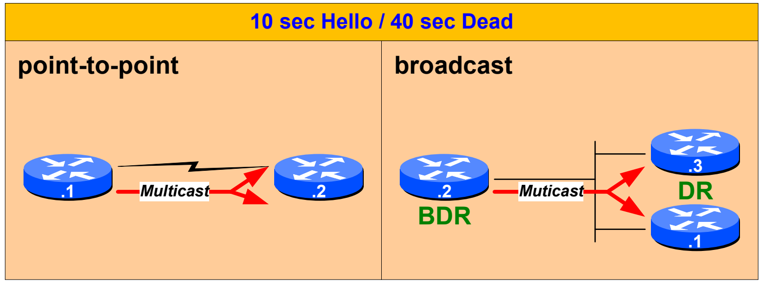
- **P-to-P**
  - No DR and BDR election. Hello sent to 224.0.0.5 (10 / 40). Neighbors always form adjacency
  - *(IF) ip ospf network point-to-point*
    Can be used on loopback interface to avertise real network and subnet. Loopback interface by default advertises /32 host address only and is set to Stub network
- **NBMA**
  - DR and BDR election. Hello sent as **unicast** (30 / 120)
  - Default on FR. Uses LSA2. Not used anymore in real scenarios
  - *(G) interface serial0/0.1 multipoint*
    This subinterface is NBMA, NOT p-t-multipoint
  - *(OSPF) neighbor <ip> [priority <id>] [poll-interval <sec>]*
    Static neighbor configuration is required (only on Hub, as spoke will learn hub's IP via unicasted Hello)
  - DR passes routes along but does not change any lookup attributes (next-hop), so static L2/L3 mapping is required between FR spokes. DMVPN does not require spoke-to-spoke mapping, because of dynamic behaviour of NHRP
  - Priority for spokes should be 0 so spokes will not become DR/BDR when hub flaps
- **Broadcast**
  - Default on ethernet
  - *(IF) ip ospf network broadcast*
    NH still not changed on Hub-Spoke FR, so L2/L3 mapping is required for spokes to communicate (with broadcast keyword)
  - DR and BDR election. Hello 10 / 40. DR and BDR use 224.0.0.6. Uses LSA2
- **Point-to-multipoint**
  - No DR and BDR election. Hello sent as **224.0.0.5** (30 / 120)
  - Networks are treated as a collection of point-to-point links. Good for DMVPN
  - Hub router changes FA to itself when passing routes between spokes
  - *(IF) ip ospf network point-to-multipoint*
    Must be set on each neighboring router, as timers are changed
  - 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
- **Demand Circuit**
  - *(IF) ip ospf demand-circuit*
    Hellos are suppressed on p2p and p2m network types. Only one side can be configured

DR/BDR Election

- DR and BDR reach full state, but DROTHER stops at 2Way with each other – no need to proceed to DBD exchange
- DR and BDR are elected per-interface. Being DR on one Eth, does not mean we are DR on other interfaces
- DR limits flooding and generates LSA2 representing shared subnet (otherwise all attached routers would describe shared subnet causing multiple LSAs with the same content)
- All routers send DBD and LSR/LSU to DR/BDR using 224.0.0.6. DR floods LSA to the segment using 224.0.0.5. BDR only listens. It takes over if flooding from DR is not heard
- When router sends own Hello and does not hear other Hellos within WAIT time (=Dead interval), it becomes DR. This is some sort of preemption, which can happen if network is misconfigured (other Hellos expire)
- When a router's interface becomes functional, it checks (Hellos) if DR and BDR is elected. If so, router accepts it regardless of own priority and router ID (no preemption), even if it was DR before link went down
- The cost from attached router to DR is the cost of that router's interface, but cost from DR to any attached router is 0
- ***(IF) ip ospf priority <#>***
  (ODPF) neighbor <ip> priority <#> (NBMA)
  Highest priority wins (default 1) or highest RID (the same priority). If set to 0 then router does not participate in election. If all routers have priority 0 neighborship is set but no adjacency
- If DR fails, BDR becomes DR and BDR is elected. When DR changes, it appears in SPF tree as an entirely new node. This causes new LSA1 and LSA2 to be originated and SPF tree rebuild on all routers in area
- **Election process:**
  - **1.** If router comes up and hears DR=0.0.0.0 in Hello (other routers also just came up) it waits Wait Time = Dead Interval, after reaching 2WAY, for other possible routers to come up. Then election process takes place
  - **2.** Calculate BDR from received Hellos. Only routers that have not declared themselves to be DR are eligible to become BDR. If one or more routers already declared themselves as BDR, the one having highest priority or router ID wins. If no routers declared BDR role, choose one from the list of all routers
    RT A: (Pri: 1); RT B: (Pri: 2); RT C: (Pri: 3) => BDR
  - **3.** Calculate DR. If one or more routers already declared themselves as DR the one having highest priority or router ID wins. If no routers declared DR role, assign DR to the router just elected as BDR
    RT A: (Pri: 1); RT B: (Pri: 2); RT C: (Pri: 3) => BDR => DR
  - **4.** If router is now DR and BDR, repeat steps 2 and 3 to select BDR from a list of remaining (non-DR) routers
    RT A: (Pri: 1); RT B: (Pri: 2) => BDR; RT C: (Pri: 3) => DR

ip ospf network	DR BDR	Hello Int	Static nghbr	Hello 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





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)

OSPF

Areas

The topology of one area is invisible to other areas. Routers in the same area have identical databases for that area

In intra-area routing, the packet is routed only using information obtained within the area

Totaly stubby
(OSPF) area <id> stub no-summary
Configured only on ABR. In addition, suppress regular LSA3 (except 0/0)

Stubby area
(OSPF) area <id> stub
Suppress LSA4 and LSA5. Generates LSA3 default with cost 1 (0/0 is not required in routing table)

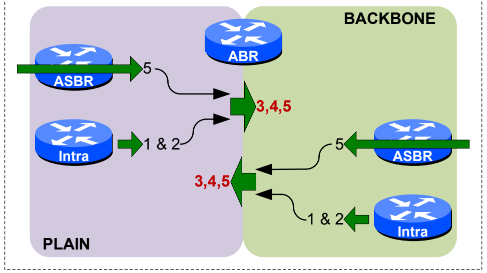
Totaly Not-so-stubby
(OSPF) area <id> nssa no-summary
Configured only on ABR. In addition suppress regular LSA3 (except generated IA 0/0)

Not-so-stubby (NSSA)
(OSPF) area <id> nssa
Suppress LSA5, but allows external LSA7 within area (translated to LSA5 by ABR). Does NOT generate default route at all

In totaly NSSA (no-summary) default route originated by ABR into area is LSA3. This insures intra-AS connectivity to the rest of the OSPF domain, as LSA3 summary route is preferred over any other default route (LSA7)

Area number is not propagated, the same area ID can be used on all areas

Plain area



What is allowed inside areas

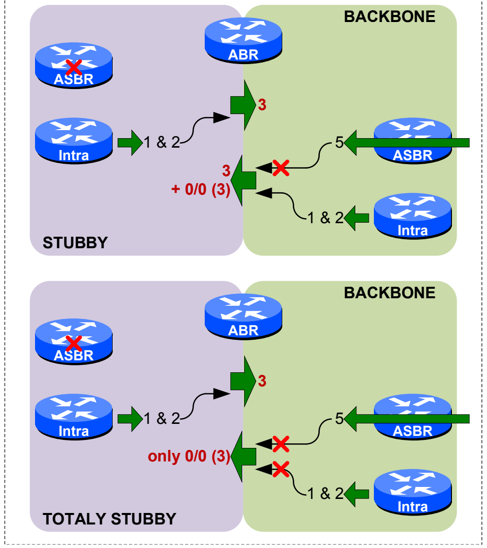
Area	1&2	3	4	5	7
Area 0	Yes	Yes	Yes	Yes	No
Regular	Yes	Yes	Yes	Yes	No
Stub	Yes	Yes	No	No	No
Totally	Yes	No*	No	No	No
NSSA	Yes	Yes	Yes	No	Yes

*Except LSA3 default route (IA)

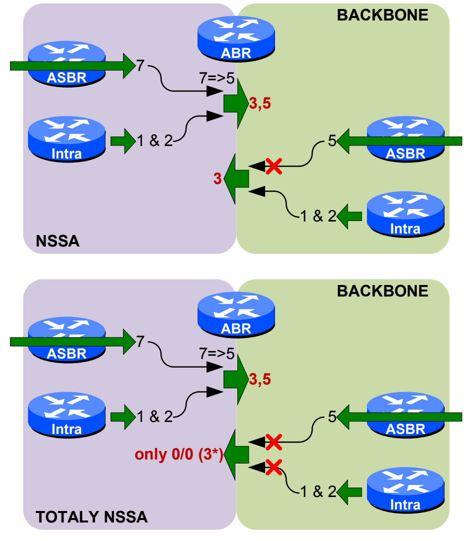
What passes between areas

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

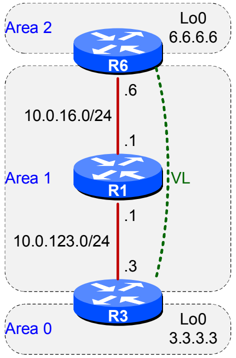
Stubby areas



Not-so-stubby areas



*) Check default information origination section for more topics on NSSA 0/0



```
R3#sh ip ospf database

            OSPF Router with ID (3.3.3.3) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         2794        0x80000007 0x001178 3
6.6.6.6         6.6.6.6         1     (DNA) 0x80000003 0x004B85 1
```

Do Not Age

```
R6#sh ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
3.3.3.3           0   FULL/  -           -        10.0.123.3      OSPF_VL1
```

No deadtime = no hellos

```
R3#sh ip ospf database router 6.6.6.6

            OSPF Router with ID (3.3.3.3) (Process ID 1)

                Router Link States (Area 0)

  Routing Bit Set on this LSA in topology Base with MTID 0
  LS age: 1 (DoNotAge)
  ...
  Area Border Router
  Number of Links: 1

    Link connected to: a Virtual Link
     (Link ID) Neighboring Router ID: 3.3.3.3
     (Link Data) Router Interface address: 10.0.16.6
      Number of MTID metrics: 0
       TOS 0 Metrics: 2
```

VL is in Area 0

New type of link

ABR connecting to real area 0

Virtual-Link

(OSPF) area <transit-area> virtual-link <RID of remote ABR>
Configured on ABRs. One must be in area 0, the other is connected to cascaded area

OSPF treats two ABRs joined by VL as if they were connected by an unnumbered point-to-point interface, so VL has no cost. It is defined to be intra-area cost between the two ABRs.

VL stays active after authentication is applied (on-demand circuit). Hello is sent over VL only once, to establish adjacency, then no hellos are sent. Disabling VL on one side is not seen on the other side (one way neighbors)

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

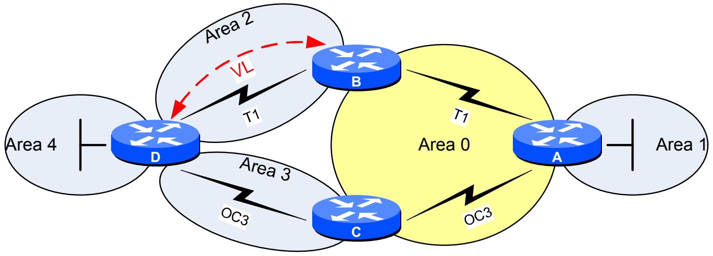
VL is an interface in area 0 (must be authenticated if area 0 is authenticated)

(OSPF) area <#> virtual-link <RID> authentication [{null | message-digest}]
Define authentication for VL: Plain text (no options), null (no authentication), or md5

(OSPF) area <#> virtual-link <RID> authentication authentication-key <string>
(OSPF) area <#> virtual-link <RID> authentication message-digest-key 1 md5 <string>
Define plain-text password or MD5 key and password

VL has no IP address, so it does not carry data traffic, only control-plane. Communication is unicatesd between real ABRs' interfaces

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



```
R1#show ip ospf database router 2.2.2.2
                           LSA1
            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Router Link States (Area 1)

  LS age: 13
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 2.2.2.2
  Advertising Router: 2.2.2.2 Router in local area
  LS Seq Number: 8000000A
  Checksum: 0x194B
  Length: 72
  Number of Links: 4

    Link connected to: a Stub Network Loopback0
     (Link ID) Network/subnet number: 2.2.2.2
     (Link Data) Network Mask: 255.255.255.255
      Number of MTID metrics: 0  Real netmask is /24 but lo0 is /32 by default
       TOS 0 Metrics: 1
                Cost of Lo0    Not only links but also routers are listed
    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 8.8.8.8 Other router's ID
     (Link Data) Router Interface address: 10.0.28.2
      Number of MTID metrics: 0                      Other router's
       TOS 0 Metrics: 1                              IP on that link

    Link connected to: a Stub Network P2P link to other router in area
     (Link ID) Network/subnet number: 10.0.28.0
     (Link Data) Network Mask: 255.255.255.0
      Number of MTID metrics: 0          ip ospf network point-to-point
       TOS 0 Metrics: 1
                           LAN with DR
    Link connected to: a Transit Network             DR IP on this segment
     (Link ID) Designated Router address: 10.0.123.1
     (Link Data) Router Interface address: 10.0.123.2
      Number of MTID metrics: 0                    Router's IP on this segment
       TOS 0 Metrics: 1 Cost from R1 to R2
```

LSA1

<table>
<tr><td>0</td><td>N</td><td>W</td><td>V</td><td>E</td><td>B</td><td>0</td><td># links (16)</td></tr>
<tr><td colspan="8">Link ID (32)</td></tr>
<tr><td colspan="8">Link Data</td></tr>
<tr><td colspan="6">Type (8)</td><td># TOS (8)</td><td>Metric (16)</td></tr>
<tr><td colspan="8">...</td></tr>
<tr><td colspan="6">TOS (8)</td><td>0</td><td>TOS Metric (16)</td></tr>
<tr><td colspan="8">Link ID (32)</td></tr>
<tr><td colspan="8">...</td></tr>
</table>

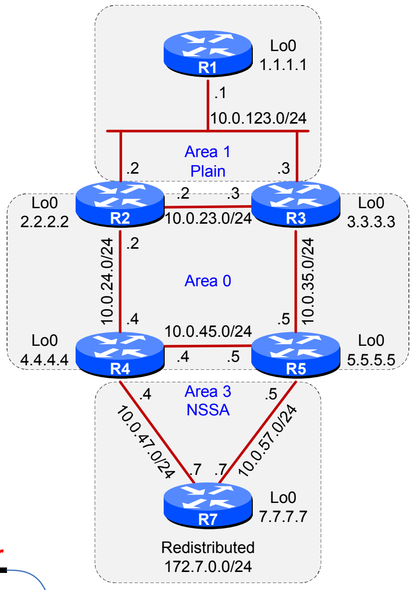
OSPF

LSA1 Router

- LSA flooded inside area only
- Router originates a LSA1 for each area that it belongs to. It describes the states of the router's links in the area
- LSA ID = Router ID originating LSA
- V: When set, the router is an endpoint of one or more fully adjacent virtual links
- E: When set, the router is an ASBR. All NSSA ABRs and NSSA ASBRs also set bit E
- B: When set, the router is an ABR
- Nt: When set, the router is an NSSA ABR that is unconditionally translating LSA7 into LSA5
- W: wild-card multicast receiver

Type	Description	Link ID
1	Point-to-point	Neighbor Router ID
2	Link to transit	Interface address of DR
3	Link to stub	IP network number
4	Virtual link	Neighbor Router ID

- „Routing Bit Set on this LSA" means that the route to this LSA1 is in routing table. If advertising router dies, all his LSAs are marked with „no routing bit set”. LSAs stay in DB untill Max LSA age passes (avoid reflooding LSAa if the router only flapped)
- OSPF advertises host routes (/32) as stub networks. Loopback interfaces are also considered stub networks and are advertised as host routes regardless of netmask, unless ***ip ospf network point-to-point*** is used
- If unnumbered interfaces are used to form adjacency, the interface address of LSA1 is set to MIB II IfIndex number
- COST: sum of all costs on links, transit networks and stub networks (local topology)
- ***show ip ospf database router***



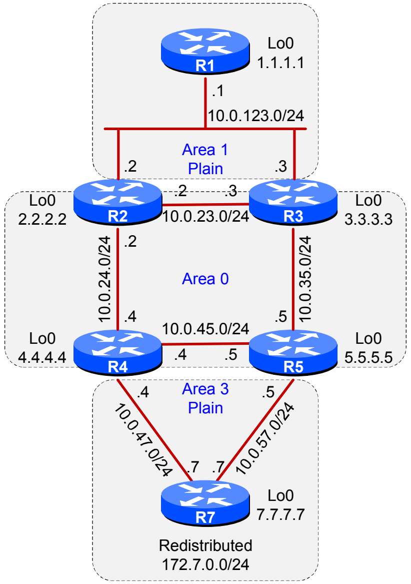
```
R1#show ip ospf database network 10.0.123.1
                           LSA2
            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Net Link States (Area 1)

  Routing Bit Set on this LSA in topology Base with MTID 0
  LS age: 1590
  Options: (No TOS-capability, DC)
  LS Type: Network Links Link ID with netmask creates a prefix
  Link State ID: 10.0.123.1 (address of Designated Router)
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000003
  Checksum: 0xF799
  Length: 36            Netmask on this subnet
  Network Mask: /24
        Attached Router: 1.1.1.1
        Attached Router: 2.2.2.2 Routers (ID) present in this segment
        Attached Router: 3.3.3.3
```

LSA2 Network

- LSA ID = DR’s interface address
- Originated only by DR
- Flooded withing area only
- Generated for every transit broadcast or NBMA network
- The DR originates the LSA only if it is fully adjacent to at least one other router on the network
- Attached router entries are the list of Router IDs of each fully adjacent routers to the DR (included). It is a pseudonode referencing to all RIDs neighboring with DR
- ***show ip ospf database network***

LSA2

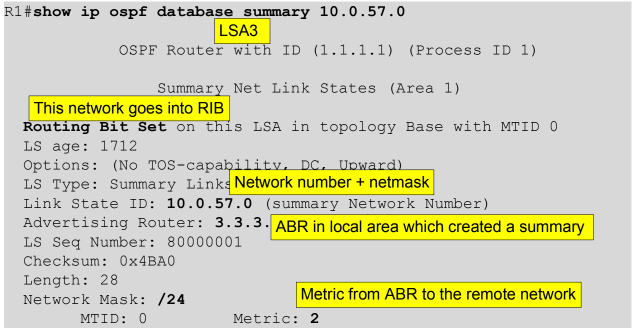
Network mask (32)
Attached router (32)
...



OSPF

LSA3 Net Summary

- LSA ID = network number
- Describes ABR's reachability to networks in other areas. Includes cost, but hides path inside original area
- LSA3 data is LSA1 & 2 as a simple subnet vector – network, netmask, and ABR's cost to reach that network
- LSA3 is flooded throughout a single area only. LSA3 generated by one ABR into area 0 is re-generated by other ABR to other areas (advertising router changes)
- When LSA1 & 2 is translated into LSA3 into area 0, LSA3 gets flooded. But, when LSA3 is to be passed from area 0 into other area, ABRs performs redistribution. So, if route in LSA3 is NOT in routing table, it is not picked up by ABR and LSA3 is not passed to that area
- Only intra-area routes are advertised into the backbone (from other areas), while both intra-area and inter-area routes are advertised into the other areas from backbone-area
- LSA3 are generated when destination is an IP network. When destination is an ASBR, LSA4 is created
- If an ABR knows multiple routes to destination within own area, it originates a single LSA3 into backbone with the lowest cost of the known 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 (LSA1 in local area)
- If a 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 by routers in other areas)
- COST: cost carried in LSA3 + cost to local ABR (from LSA1)
  - Cost from R1 to 10.0.57.0/24 is 2 (in LSA3) + 1 (LSA1 from R3)
- ***show ip ospf database summary***
- ***show ip ospf border-router***
  Shows ABRs and ASBRs from whole routing domain, even from different areas


LSA3
This network goes into RIB
Network number + netmask
ABR in local area which created a summary
Metric from ABR to the remote network

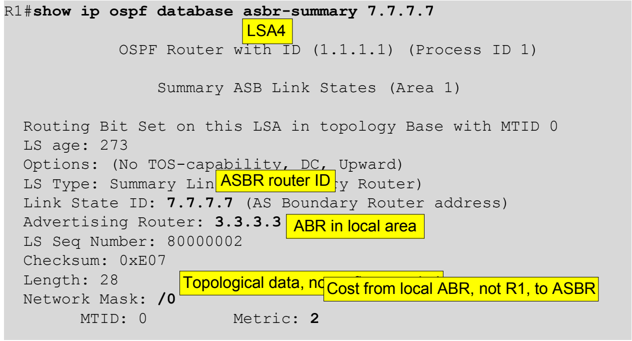
```
R1#show ip ospf database summary 10.0.57.0

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Summary Net Link States (Area 1)

  Routing Bit Set on this LSA in topology Base with MTID 0
  LS age: 1712
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Links
  Link State ID: 10.0.57.0 (summary Network Number)
  Advertising Router: 3.3.3.
  LS Seq Number: 80000001
  Checksum: 0x4BA0
  Length: 28
  Network Mask: /24
        MTID: 0         Metric: 2
```

LSA3/4

Network mask (32)	
0	Metric (24)
TOS (8)	TOS Metric (24)
...	

LSA4 ASBR Summary

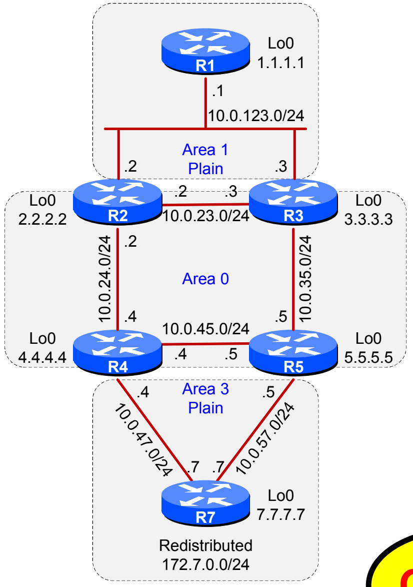
- LSA ID = ASBR RID
- ASBR generates LSA1 with special characteristics (E-bit set) - *AS Boundary Router* displayed in LSA1
- LSA4 is generated by ABR into backbone area and regenrated by another ABR into non-backbone area
- No LSA4 in original area
- Routers which receive external routes inside original area, already know how to get to the ASBR (LSA1 is generated by ASBR)
- The LSA4 does not contain information about reachable subnets. It is just a topological component that is necessary to find a way to ASBR (router ID). The LSA5 depends on LSA4, but NOT LSA7 translated into LSA5
- When routers inside other areas receive LSA5, advertising router for that route points to ASBR RID (do not confuse with prefix, as router ID is IP-address-alike). Routers in other areas have no idea how to get to that RID (topologically), so they need the LSA4
- Cost in LSA4 is from local ABR to remote ASBR. Local cost from inside router to ABR must be added to caluclations (LSA1). Cost in LSA4 generated int non-backbone area is cumulative (cost from original ABR to ASBR + cost from non-backbone ABR to original ABR)
- Cost in LSA4 from R1: 1 (R3 to R5) + 1 (R5 to R7) = 2
- ***show ip ospf database asbr-summary***



```
R1#show ip ospf database asbr-summary 7.7.7.7

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Summary ASB Link States (Area 1)

  Routing Bit Set on this LSA in topology Base with MTID 0
  LS age: 273
  Options: (No TOS-capability, DC, Upward)
  LS Type: Summary Lin         y Router)
  Link State ID: 7.7.7.7 (AS Boundary Router address)
  Advertising Router: 3.3.3.3
  LS Seq Number: 80000002
  Checksum: 0xE07
  Length: 28
  Network Mask: /0
        MTID: 0         Metric: 2
```

Other LSAs

- LSA6: Group membership
  - ***(OSPF) 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)



LSA5/7

Nertwork mask (32)		
E	0	Metric (24)
Forwarding address (32)		
Tag (32)		
E	TOS (7)	TOS Metric (24)
Forwarding address (32)		
Tag (32)		
...		

OSPF

LSA5
AS External

- LSA ID = external network number
- E: Type of metric, if set, the metric is a Type 2 (default), otherwise it's Type 1
- LSA5 is created by ASBR, and is flooded unaltered throughout the entire domain
- Default cost of routes redistributed into OSPF is 20
- Internal cost inside LSA5 is not altered on the path. Only SPF calculations are different for E1 and E2
- If FA is set to 0.0.0.0, packets should be sent to the ASBR itself (NH for redistributed subnet is not a native part of OSPF). Searching for ASBR, select the routing table entry with the least cost. When there are multiple least costs, the entry from the largest OSPF Area ID
- If FA is non-zero, it must be in routing table reachable natively by OSPF (cannot be external route). Non-zero FA is set when ASBR's external link pointing to NH is included with network statement
- If an ASBR within a non-backbone area advertises an external route it is preferred over external routes advertised by ASBRs in other areas regardless of metric
- For LSA3 and LSA5 the LS ID may additionally have one or more of the destination network's "host" bits set. For ex. when originating an LSA5 for the network 10.0.0.0 with mask of 255.0.0.0, the Link State ID can be set to anything in the range 10.0.0.0 through 10.255.255.255 inclusive. This allows a router to originate separate LSAs for two networks having the same address but different masks
- If local routers select exit point based on the external metric (E2) they perform "cold potato" routing. If local path is included in calculations (E1) then it's „hot potato" routing – more optimal exit path
- E1 cost = 20: redistributed (LSA5) + 1: cost to closest ABR (R3/LSA1) + 2: cost from local ABR to remote ASBR = 23
- E2 cost = 20: redistributed (LSA5)
  - **1.** If two ASBRs redistribute the same prefix, the one with lower redistributet metric is choosen
  - **2.** If redistributed metrics are the same, lower cost to ASBR is choosen (forward metric)
  - **3.** If forward metrics are the same, ECMP is used
- ***show ip ospf database external***

```
R1#show ip ospf database external 172.7.0.0

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Type-5 AS External Link States

  Routing Bit Set on this LSA in topology Base with MTID 0
  LS age: 15
  Options: (No TOS-capability, DC, Upward)
  LS Type: AS Exter
  Link State ID: 172.7.0.0 (External Network Number )
  Advertising Router: 7.7.7.7
  LS Seq Number: 80000001
  Checksum: 0xFBD4
  Length: 36
  Network Mask: /24
        Metric Type: 2 (Larger than any link state path)
        MTID: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 0
```

LSA5 E2

Network number + netmask goes into RIB

ASBR router ID in original area

Metric not changed along the path

FA 0.0.0.0 means use advertising router ID

```
R1#show ip route 172.7.0.0 255.255.255.0
Routing entry for 172.7.0.0/24
  Known via "ospf 1", distance 110, metric 20, type extern 2, forward metric 3
```

E2 metric

Cost from local router to ASBR

```
R1#show ip ospf border-routers

Codes: i - Intra-area route, I - Inter-area route

I 5.5.5.5 [2] via 10.0.123.3, GigabitEthernet0/0, ASBR, Area 1, SPF 6
I 5.5.5.5 [3] via 10.0.123.2, GigabitEthernet0/0, ASBR, Area 1, SPF 6
i 2.2.2.2 [1] via 10.0.123.2, GigabitEthernet0/0, ABR, Area 1, SPF 6
i 3.3.3.3 [1] via 10.0.123.3, GigabitEthernet0/0, ABR, Area 1, SPF 6
```

Internal costs to ABRs/ASBRs

```
R1#show ip ospf database external 172.7.0.0

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Type-5 AS External Link States

  Routing Bit Set on this LSA in topology Base with MTID 0
  LS age: 170
  Options: (No TOS-capability, DC, Upward)
  LS Type: AS Exter
  Link State ID: 172.7.0.0 (External Network Number )
  Advertising Router: 7.7.7.7
  LS Seq Number: 80000006
  Checksum: 0x6EDD
  Length: 36
  Network Mask: /24
        Metric Type: 1 (Comparable directly to link state metric)
        MTID: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 0
```

LSA5 E1

Network number + netmask goes into RIB

ASBR router ID in original area

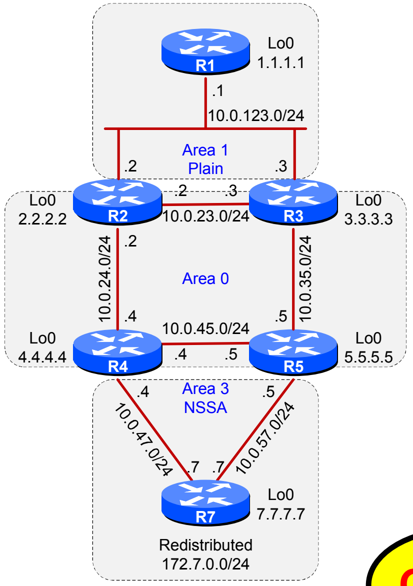
Metric inside LSA5 is not changed along the path

FA 0.0.0.0 means use advertising router ID

```
R1#show ip rou 172.7.0.0 255.255.255.0
Routing entry for 172.7.0.0/24
  Known via "ospf 1", distance 110, metric 23, type extern 1
```

E1 cumuative metric

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



OSPF

LSA7 NSSA External

- LSID = external network number
- Forwarding address: **1)** highest IP on loopback interfaces, **2)** highest IP on physical interface. OSPF must be enabled on the interface to be considered for FA. The FA MUST be reachable in the whole OSPF domains as OSPF route, not from other protocol
- Forwarding address is preserved during LSA7=>LSA5 translation, so no LSA4 is required to reach translated LSA7 route. NH is taken from RIB
- Flooded only within the not-so-stubby area in which it was originated. Blocked by ABR and Translated into LSA5. If many ABRs exist only the one with highest Router ID does the translation
- FA in translated LSA5 is set to original ASBR router, not ABR (0.0.0.0), so optimal path can be selected regardless of which ABR performed translation. Path is selected based on forwarding metric to ASBR, not to ABS which did the translation
- LSA format is exactly the same as for LSA5, except of meaning of FA and P-bit (OSPF hello header)
- NP: If set, translate LSA7 into LSA5 and flood it throughout the other areas (FA must be then non-zero). If not set, then no translation takes place, and the prefix will not be advertised outside NSSA
- NP-bit is always set by default in Hello. To stop translation ***summary-address*** with ***not-advertise*** can be used on ABR ONLY
- ***(OSPF) area <id> nssa no-redistribution***
  Used when an NSSA ABR is also an ASBR. LSA7 into NSSA is suppressed, but routes are still redistributed to plain and backbone areas. When an NSSA ABR originates both LSA5 and LSA7 for the same network, and P-bit is set (there is no way to clear P-bit) it may be translated into LSA5 by another NSSA ABR causing suboptimal paths. LSA with P-bit set is preferred over one with the P-bit clear. If the P-bit settings are the same, the LSA with the higher router ID is preferred.
- Default (0/0) originated by an NSSA ABR is never translated into a LSA5, however, a Type-7 default LSA originated by internal ASBR may be translated into LSA5
- ***(OSPF) area <id> nssa translate type7 suppress-fa***
  Configured on ABR. Sets FA to 0.0.0.0 (ABR becomes FA). This feature is noncompliant with RFC 1587 (caution!). Helpfull if area summarization is used with ***no-advertise*** keyword, so area's intra-area routes are filtered, and FA for LSA5 becomes unavailable. Non-reachable next-hop means no route in RIB.
- ***(OSPF) area <id> nssa translate type7 always***
  Force ABR to win election if there is another ABR with higher Router ID
- NSSA ABR converts LSA7 into LSA5 and inject it into the backbone, so it becomes an ASBR (E-bit set in LSA1 in area0), so *AS Boundary Router* and *Area Border Router* are displayed in LSA1
- ***show ip ospf database nssa-external***

LSA5/7

Nertwork mask (32)		
E	0	Metric (24)
Forwarding address (32)		
Tag (32)		
E	TOS (7)	TOS Metric (24)
Forwarding address (32)		
Tag (32)		
...		

```
R7#show ip ospf database nssa-external

            OSPF Router with ID (7.7.7.7) (Process ID 1)

                Type-7 AS External Link States (Area 3)

  LS age: 175
  Options: (No TOS-capability, Type 7/5 translation, DC, Upward)
  LS Type: AS External I
  Link State ID: 172.7.0.0 (External Network Number )
  Advertising Router: 7.7.7.7
  LS Seq Number: 80000005
  Checksum: 0xBEE7
  Length: 36
  Network Mask: /24
        Metric Type: 2 (Larger than any link state path)
        MTID: 0
        Metric: 20
        Forward Address: 7.7.7.7
        External Route Tag: 0
```

Only on routers inside NSSA

LSA7 N2

P-bit set

Network number + netmask goes into RIB

ASBR in local area (router-ID)

Metric not changed along the path

FA set to highest loopback or physical interface

```
R1#show ip ospf database external 172.7.0.0

            OSPF Router with ID (1.1.1.1) (Process ID 1)

                Type-5 AS External Link States

  Routing Bit Set on this LSA in topology Base with MTID 0
  LS age: 14
  Options: (No TOS-capability, DC, Upward)
  LS Type: AS Extern
  Link State ID: 172.7.0.0 (External Network Number )
  Advertising Router: 5.5.5.5
  LS Seq Number: 80000003
  Checksum: 0x9327
  Length: 36
  Network Mask: /24
        Metric Type: 2 (Larger than any link state path)
        MTID: 0
        Metric: 20
        Forward Address: 7.7.7.7
        External Route Tag: 0
```

Translated LSA5 E2

Network number + netmask goes into RIB

ABR doing translation 7 > 5

Metric not changed along the path

FA preserved by ABR doing translation

```
R1#sh ip route 172.7.0.0 255.255.255.0
Routing entry for 172.7.0.0/24
  Known via "ospf 1", distance 110, metric 20, type extern 2, forward metric 4
```

N2 metric

Cost from local router to ASBR

```
R1#sh ip route 7.7.7.7
Routing entry for 7.7.7.7/32
  Known via "ospf 1", distance 110, metric 4, type inter area
```

Forward metric for LSA5 with FA set. NH is ASBR's loopback (additional cost of 1)

OSPF

Distance

Routes learned from two different processes cannot be compared (all routes in one process are completely different than in another process). First come, first served. AD should be used to differentiate those routes

(OSPF) distance ospf {external | inter-area | intra-area} <ad>
Change AD for specific routes.

(OSPF) distance <ad> <source> <source wildcard> <prefix acl>
Change AD for specific prefixes (ACL) received from specific sources. Source is a ROUTER ID of a outer which originated LSA, not neighbor's IP address

Cost

Path selection preference (for the same prefix, regardless of the cost value): Intra-Area (O), Inter-Area (O IA), External Type 1 (E1), NSSA Type 1 (N1), External Type 2 (E2), NSSA Type 2 (N2)

E1/N1 or E2/N2 route selection is used if Forward Metric is the same, otherwise better Forward Metric to the destination (ASBR) always wins, regardless of route type. Type 1 is ALWAYS better than Type 2 regardless of the Forward Metric

(OSPF) auto-cost reference-bandwidth <bw in Mbps>
Default reference: 100 Mbps / intf BW (FE and faster intf. get 1). Should be the same on all routers

(IF) ip ospf cost <cost>

(OSPF) neighbor <ip> cost <cost>
Valid only for point-to-multipoint and point-to-multipoint non-broadcast networks (spokes with different CIRs)

(OSPF) area <id> default-cost <cost>
Set default cost for redistributed routes (default is 1 for BGP, 20 for other routing protocols, and 0 for connected and static routes), but also for default route originated into area

Do NOT change bandwidth to manipulate OSPF cost, as BW is also used by QoS, EIGRP, etc

Summary

(OSPF) summary-address <prefix> <mask> [no-advertise] [tag <tag>] [nssa-only]
External routes (**LSA5 and LSA7**) can be summarized only on ASBR, which does redistribution. Cost is taken from smallest cost of component routes. The **not-advertise** means no advertising to any area, so in effect, discard summary route is not generated and all covered routes are filtered from database and advertisement. To clear P-bit inside NSSA use ***nssa-only*** option

Summarization on NSSA ASBR takes FA from the best smaller redistributed route with lowest metric

(OSPF) area <id> range <prefix> <mask> [advertise | not-advertise] [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. If ***not-advertise*** is used LSA3 is suppressed (no discard route), and the component routes are filtered from database

(OSPF) discard-route [external [<AD>]] [internal [<AD>]]
Summarized routes automaticaly create static Null0 route to prevent loops. By default AD for external routes is 254, and 110 for internal routes

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

Default route

You cannot redistribute a default route from other routing protocols. OSPF treats it as a special route

If regular router originates 0/0 it becomes an ASBR. If ABR originates 0/0 it is NOT an ASBR

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

(OSPF) default-information originate [always] [metric <#>] [metric-type {1 | 2}] [route-map <name>]
Default originated into all attached plain areas. Injected as LSA5 (type-1 or type-2). Default must be in routing table, unless ***always*** is defined. Metric is 1 by default. Default route can be originated conditionaly with route-map

Stubby and totaly stubby areas automaticaly generate 0/0 (ABR) with cost 1. Default is not required to be present in routing table on ABR

Totaly NSSA automatically generates LSA3 0/0 with cost 1

(OSPF) area <id> nssa default-information-originate [metric <#>] [metric-type {1 | 2}]
Generate N2 default route into NSSA area. Default route does NOT have to be in routing table. Metric is 1

(OSPF) area <id> nssa no-summary default-information-originate [metric <#>] [metric-type {1 | 2}]
Overrides ***no-summary*** LSA3 default route generation and generates N2 default route. Metric is 1

If metric is the same then forward metric is used to select 0/0

Redistribution

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)

filter-list

Configured on ABR at the point where LSA3 would be created. Filters **ONLY LSA3**, which is a plain prefix, so can be filtered on ABR. There is a distance-vector behavior between areas

(OSPF) area <#> filter-list prefix <name> {in | out}
Prefix list defines what is allowed, NOT filtered!

in – into area <#>. Prefix is allowed from area 0 into area <#> only if prefix-list matches it exactly, regardless whether it is a plain LSA3 generated by other ABR or LSA3s aggregated with area range

out – into area 0. Prefix is allowed from area <#> into area 0, if prefix-list matches it exactly, however, if area range is configured on that ABR, aggregated prefix is allowed if prefix-list matches at least one of more specific prefixes (although the smaller prefix is not allowed – it gets aggregated)

distribute-list

Filters („in" means into routing table) ANY **LSA3 IA** 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. LSA5 cannot be filtered on regular ABRs, as it is flooded through whole domain

(OSPF) distribute-list <acl> {in [<if>] | out [{<if> | <protocol>}]}
Only routes matched by ACL will be injected into RIB or sent to a neighbor. **Note:** if extended ACL is used, source part matches Router ID of route originator, and destination part matches subnets allowed

(OSPF) distribute-list gateway <prefix list> {in [<if>] | out [{<if> | <protocol>}]}
Allows only prefixes received from neighbor listed in gateway prefix list. The gateway prefix list defines neighbor's interface IP address, NOT router ID

(OSPF) distribute-list prefix <list> [gateway <prefix list>] {in [<if>] | out [{<if> | <protocol>}]}
Allows only specific prefixes defined with prefix list, received from neighbor listed in gateway prefix list. The gateway prefix list defines neighbor's interface IP address, NOT router ID

(OSPF) distribute-list route-map <name> {in [<if>] | out [{<if> | <protocol>}]}
You can filter inbound prefixes based on tag, next-hop, etc

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

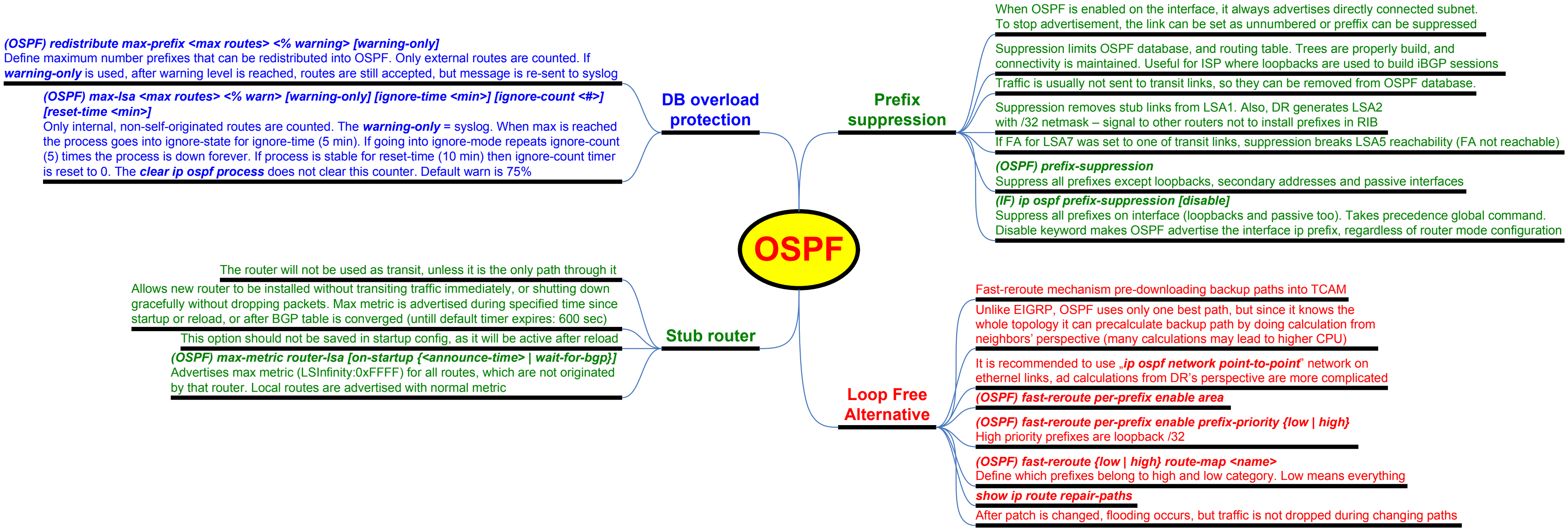
If route-map is used, route can be matched with „***match ip route-source <acl>***" matching RID, not NH (same when using ***gateway***)

Database filtering

All outgoing LSAs are filtered.

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

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



OSPF

DB overload protection

- ***(OSPF) redistribute max-prefix <max routes> <% warning> [warning-only]***
  Define maximum number prefixes that can be redistributed into OSPF. Only external routes are counted. If ***warning-only*** is used, after warning level is reached, routes are still accepted, but message is re-sent to syslog
- ***(OSPF) max-lsa <max routes> <% warn> [warning-only] [ignore-time <min>] [ignore-count <#>] [reset-time <min>]***
  Only internal, non-self-originated routes are counted. The ***warning-only*** = syslog. 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 the process is down forever. If process is stable for reset-time (10 min) then ignore-count timer is reset to 0. The ***clear ip ospf process*** does not clear this counter. Default warn is 75%

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
- Suppression limits OSPF database, and routing table. Trees are properly build, and connectivity is maintained. Useful for ISP where loopbacks are used to build iBGP sessions
- Traffic is usually not sent to transit links, so they can be removed from OSPF database.
- Suppression removes stub links from LSA1. Also, DR generates LSA2 with /32 netmask – signal to other routers not to install prefixes in RIB
- If FA for LSA7 was set to one of transit links, suppression breaks LSA5 reachability (FA not reachable)
- ***(OSPF) prefix-suppression***
  Suppress all prefixes except loopbacks, secondary addresses and passive interfaces
- ***(IF) ip ospf prefix-suppression [disable]***
  Suppress all prefixes on interface (loopbacks and passive too). Takes precedence global command. Disable keyword makes OSPF advertise the interface ip prefix, regardless of router mode configuration

Stub router

- The router will not be used as transit, unless it is the only path through it
- Allows new router to be installed without transiting traffic immediately, or shutting down gracefully without dropping packets. Max metric is advertised during specified time since startup or reload, or after BGP table is converged (untill default timer expires: 600 sec)
- This option should not be saved in startup config, as it will be active after reload
- ***(OSPF) max-metric router-lsa [on-startup {<announce-time> | wait-for-bgp}]***
  Advertises max metric (LSInfinity:0xFFFF) for all routes, which are not originated by that router. Local routes are advertised with normal metric

Loop Free Alternative

- Fast-reroute mechanism pre-downloading backup paths into TCAM
- Unlike EIGRP, OSPF uses only one best path, but since it knows the whole topology it can precalculate backup path by doing calculation from neighbors' perspective (many calculations may lead to higher CPU)
- It is recommended to use „***ip ospf network point-to-point***" network on ethernel links, ad calculations from DR's perspective are more complicated
- ***(OSPF) fast-reroute per-prefix enable area***
- ***(OSPF) fast-reroute per-prefix enable prefix-priority {low | high}***
  High priority prefixes are loopback /32
- ***(OSPF) fast-reroute {low | high} route-map <name>***
  Define which prefixes belong to high and low category. Low means everything
- ***show ip route repair-paths***
- After patch is changed, flooding occurs, but traffic is not dropped during changing paths

OSPFv3 Header (24B)

Version (8)	Type (8)	Packet length (16)	
Router ID (32)			
Area ID (32)			
Checksum (16)		Instance (8)	0

OSPF

OSPFv3

- Multiprotocol. Works for IPv4 and IPv6. One control plane
  - OSPFv3 can be used only for IPv4 (easy migration to IPv6 in the future, one protocol)
  - IPv6 addresses are FF02::5 All OSPF hosts; FF02::6 All DR
- v2 and v3 have different SPFs. They are not compatible. Operations and logic are basically the same
- 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
- ***(IF) ipv6 ospf <id> area <area> [instance <0-255>]***
  IPv6 only. 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
- ***(IF) ospfv3 <id> [ipv4 | ipv6] area <id>***
  Multiptotocol approach for configuring OSPFv3
  - ***show ospfv3 ...***
- Link-Local address are used for adjacency (source of hello packets). On virtual links, a global scope IPv6 address must be used as the source address
- LSA1 and LSA2 only represent router's information for SPF. Flooded only if pertinent to SPF algorithm changes. If a prefix changes, it 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 router's link-local address to routers attached to the link
  - Provides a list of IPv4/IPv6 prefixes associated with the link
  - Provides Option bits
- **Intra-Area Prefix LSA** – flooded through area when a link or its prefix changes. Router LSA and Network LSA does not contain networks, they are only used to build topology
- Authentication (AH or ESP) and encryption (ESP) in OSPFv3 relies on underlying IPSec (no native authentication). It creates local crypto tunnel with identities for only OSPF traffic. No ISAKMP, 128 bit keys must be defined manually
- If authentication is configured you cannot add encryption. If encryption is configured it also uses authentication
- ***(IF) ipv6 ospf encryption ipsec spi <id> esp {des | 3des | aes-cbc} <key len> <encr key> {sha1 | md5} <auth key>***
- ***(IF) ipv6 ospf authentication ipsec spi <id> {sha1 | md5} <key>***
  (OSPF) area 0 authentication ipsec spi <id> {sha1 | md5} <key>
- ***show crypto ipsec sa ipv6***
- ***show ipv6 ospf database router adv-router <router-id>***
  Database does not show LSA IDs, but advertising router ID

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

```
R3#show ipv6 ospf database inter-area prefix adv-router 1.1.1.1

            OSPFv3 Router with ID (3.3.3.3) (Process ID 1)

                Inter Area Prefix Link States (Area 0)

  Routing Bit Set on this LSA
  LS age: 54
  LS Type: Inter Area Prefix Links
  Link State ID: 0
  Advertising Router: 1.1.1.1
  LS Seq Number: 80000001
  Checksum: 0xDCBE
  Length: 44
  Metric: 0
  Prefix Address: 2002:CC1E:1::1
  Prefix Length: 128, Options: None
```

Type-9 Intra-area LSA

Prefix information

```
R3#show ip ospf database

Link ID         ADV Router      Age         Seq#       Checksum Link count
3.3.3.3         3.3.3.3         90          0x80000001 0x009CFF 3
```

OSPFv2 Link ID

```
R3#show ip ospf database router 3.3.3.3

  [...]
  Link State ID: 3.3.3.3
  Advertising Router: 3.3.3.3
  [...]

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.0.35.0
     (Link Data) Network Mask: 255.255.255.0
      Number of MTID metrics: 0
       TOS 0 Metrics: 1
```

Prefix information

```
R3#show ipv6 ospf database

ADV Router                Age  Seq#         Fragment ID  Link count  Bits
 3.3.3.3                  7    0x80000001   0            0           None
```

OSPFv3 Advertising router ID

```
R3#show ipv6 ospf database router adv-router 3.3.3.3

            OSPFv3 Router with ID (3.3.3.3) (Process ID 1)

                Router Link States (Area 0)

  LS age: 43
  Options: (V6-Bit, E-Bit, R-bit, DC-Bit)
  LS Type: Router Links
  Link State ID: 0
  Advertising Router: 3.3.3.3
  LS Seq Number: 80000002
  Checksum: 0x22BD
  Length: 40
  Number of Links: 1

    Link connected to: a Transit Network
      Link Metric: 1
      Local Interface ID: 3
      Neighbor (DR) Interface ID: 3
      Neighbor (DR) Router ID: 3.3.3.3
```

No prefix information, only topology

```
R3#show ipv6 ospf database link adv-router 3.3.3.3

            OSPFv3 Router with ID (3.3.3.3) (Process ID 1)

                Link (Type-8) Link States (Area 0)

  LS age: 382
  Options: (V6-Bit, E-Bit, R-bit, DC-Bit)
  LS Type: Link-LSA (Interface: GigabitEthernet0/0)
  Link State ID: 3 (Interface ID)
  Advertising Router: 3.3.3.3
  LS Seq Number: 80000002
  Checksum: 0x7447
  Length: 68
  Router Priority: 1
  Link Local Address: FE80::C803:BFF:FE38:8
  Number of Prefixes: 2
  Prefix Address: 2002:CC1E::
  Prefix Length: 64, Options: None
  Prefix Address: 2002:CC1E::
  Prefix Length: 64, Options: None
```

Type-8 Link LSA

Prefix information

IS-IS

Features

- AD 115
- Only one ISIS process can run on a router for IP, but multiple for CLNS
- Runs directly over Layer 2 (0xFEFE), does not require L3. Neighbors exchange PDUs
- SAP is the transport (DSAP 1 byte, SSAP 1 byte, Control 1 byte). Default MTU is 1497
- Encodes the data in TLVs (Type, Length, Value)
- ***(G) router isis [<tag>]***
- ***(ISIS) protocol shutdown***
  (IF) isis protocol shutdown
  Administrively shutdown ISIS on an interface or globaly without removing configuration
- ***(ISIS) hostname dynamic***
  The router-name-to-system-ID mapping information is flooded with special TLV. If router stops flooding this information it is kept by other routers for 60 minuts
- ***show clns [protocol]***

NET

AFI (1)	Area		
Area (1 - 13)		System ID (6)	N-SEL (1)

Max 20 bytes

- ***(ISIS) net <id>***
- NSAP – Network Service Access Point - the address at which the network service is accessible. One per router (globally for all interfaces). Max 20 bytes
- NET – Network Entity Title – the address of the entity. It's an NSAP with N-SEL=0
- AFI – Authority and Format Identifier. The most common used: 39 (Country), 47 (International), 49 (Private).
- N-SEL – Network Selector – always 0 for a router, and non-sero for pseudonodes (similiar to a TCP port)
- System ID is usually transformed loopback address. 192.168.10.1 => 1921.6801.0001. Level 1 ID must be unique among all L1 routers in the same area. Level 2 ID must be unique among all routers in the domain
- ***(ISIS) max-area-addresses <#>***
  Multiple NETs are supported. Default is 3

Areas

- There can be multiple Level 1 areas interconnected by only one, contiguous Level 2 backbone
- Separate adjacencies for each level with independent SPFs. Area address must match to form an adjacency
- L1 (plain area) and L2 (backbone) hierarchy. L2 MUST be contiguous, no virtual-links
- L1 routers know topology of the own area only (stub area). L1L2 routers advertise within L2 domain all routes learned from L1 and L2 peers
- ***show isis topology***
- ***(ISIS) is-type {level-1 | level-1-2 | level-2-only}***
  (IF) isis circuit-type {level-1 | level-1-2 | level-2-only}
  Defined globaly for all enabled interfaces. Interface takes precedence. Default is level-1-2

Metric

- Metric is simply cumulative
- Narrow: max link metric is 63 (6 bits), max path metric is 1023
- Wide
  - Max link metric is 2^24 – 1, max path metric is 2^32 – 2^25
  - Extended IS Reachability TLV 22 (24bit) and Extended IP Reachability TLV 135 (32bit)
  - ***(ISIS) metric-style wide***
    Must be set on all routers (recognize TLV)
  - ***(ISIS) metric-style [{narrow | wide}] transition***
    Advertise and accept both types of metrics
- ***(ISIS) metric <#>***
  Default metric is 10 for each active interface, and 0 for passive
- ***(IF) isis metric {<#> | maximum} [{level-1 | level-2}]***
  If maximum is used, the link is not used in SPF calculations as a best path
- Path selection
  - **1.** Level 1 is preferred over Level 2
  - **2.** Internal metric-type is preferred over external metric-type
  - **3.** Lowest metric
  - **4.** Multipathing – up to 6 paths

Neighbors

- ***(IF) ip router isis [<tag>]***
  Sessions can be established ONLY between the same levels and the same Area ID (NET)
- ***(ISIS) passive-interface {<if> | default}***
  Passive interface removes ***ip router isis*** from that interface
- Hello Packets (IIH) are used to form adjacencies. Different on point-to-point links and LAN
- ***(ISIS) no hello padding [{multi-point | point-to-point}] [always]***
  (IF) no isis hello padding [always]
  IS-IS by default pads the Hellos to the full interface MTU size to detect MTU mismatches. Even if disabled, few hellos are sent with padding, unless hidden ***always*** is used
- Only point-to-point and broadcast networks are available
- ***(IF) isis network point-to-point***
  Set on Eth interface where only 2 routers exist, no DIS election
- DIS
  - Pseudonode describes the LAN (like DR in OSPF). It is created by a Designated Router (DIS). No backup DIS. Separate for L1 and L2.
  - Election is preemptive. New router with better priority takes over (new election) and generates new CSNPs. No backup DIS
  - ***(IF) isis priority <0-127> [{level-1 | level-2}]***
    Default is 64. Higher is better. If the same, MAC or DLCI is used. System-ID is a final tie-breaker. If priority is set to 0, the router does not participate in election
- Adjacency filter
  - ***(G) clns filter-set <name> {permit | deny}***
    Use * as a wildcard in place of each NET number
  - ***(IF) isis adjacency-filter <name> [match-all]***
- ***show clns {interface | neighbor}***
- ***show isis {neighbor | hostname}***

```
R1#show clns neighbors
System Id       Interface   SNPA                State  Holdtime  Type Protocol
R2              Gi0/0       ca02.3ac0.0008      Up     8         L1L2 IS-IS
R4              Gi1/0       ca04.4a2c.001c      Up     295       IS   ES-IS
R1#show isis neighbors
System Id       Type Interface   IP Address      State Holdtime Circuit Id
R2              L1   Gi0/0       10.0.123.2      UP    8        R2.01
R2              L2   Gi0/0       10.0.123.2      UP    9        R2.01
R4              L1   Gi1/0       10.0.24.4       UP    27       00
```

If ES-IS, check MTU or area (L1/L2)

p2p

```
R1#show clns interface
GigabitEthernet0/0 is up, line protocol is up
  Checksums enabled, MTU 1497, Encapsulation SAP
[...]
  Routing Protocol: IS-IS
    Circuit Type: level-1-2
    Interface number 0x0, local circuit ID 0x1
    Level-1 Metric: 10, Priority: 64, Circuit ID: R2.01
    DR ID: R2.01
[...]
```

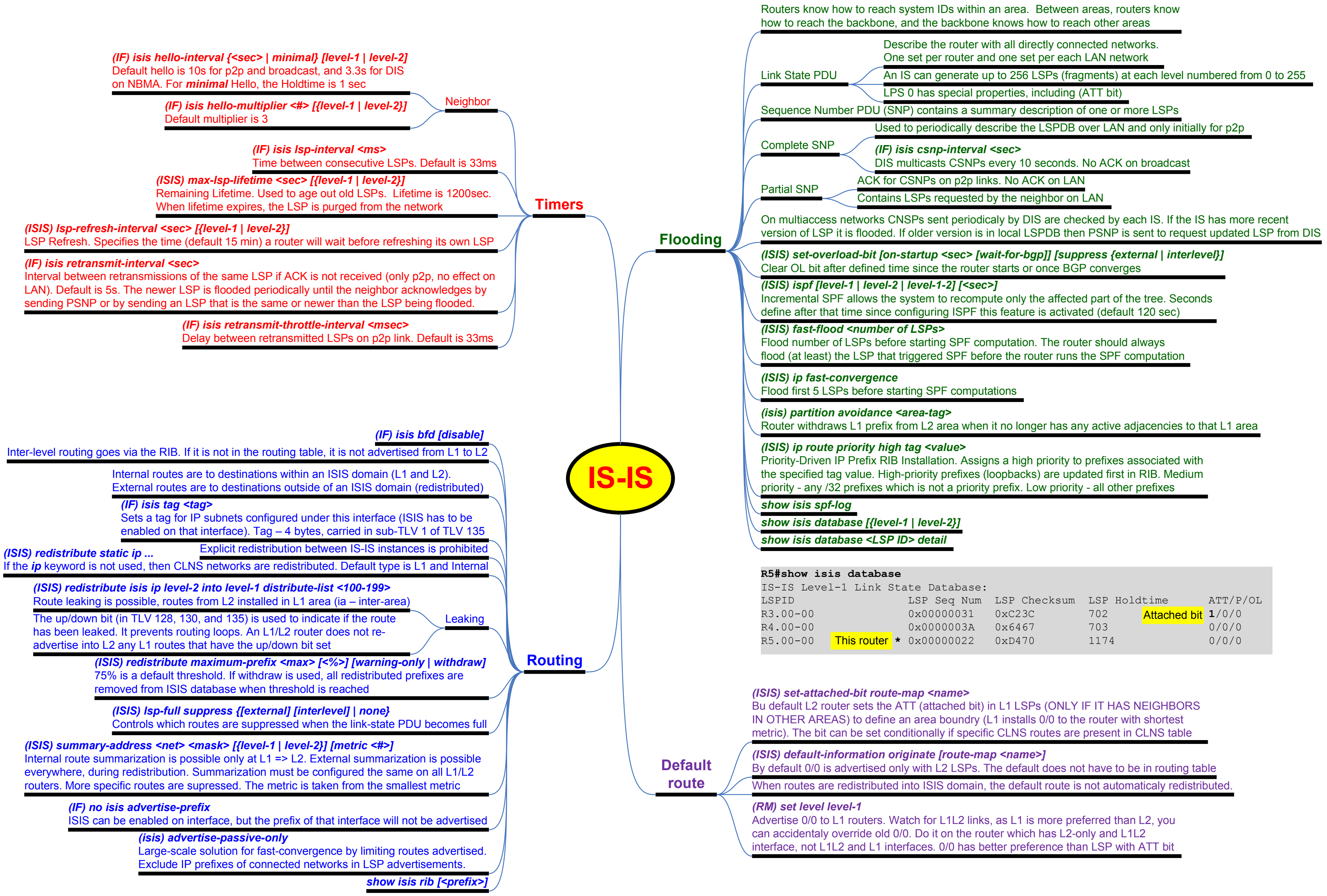
SAP encapsulation is 3 bytes

Default setting

For DIS election

Authentication

- Authentication applied to an interface authenticates Hello PDUs, but when applied to the ISIS globally, authenticates also LSPs, CSNPs, and PSNPs
- ***(ISIS) isis authentication mode {text | md5} [{level-1 | level-2}]***
  (IF) isis authentication mode {text | md5} [{level-1 | level-2}]
- ***(ISIS) isis authentication key-chain <name> [{level-1 | level-2}]***
  (IF) isis authentication key-chain <name> [{level-1 | level-2}]
- Old
  - ***(IF) isis password <text>***
    Plain text password used for Hello adjacency
  - ***(ISIS) area-password <password>***
    Level-1 password. Set in LSPs, CSNPs, and PSNPs
  - ***(ISIS) domain-password password [authenticate snp {validate | send-only}]***
    Level-2 password. Set in LSPs, CSNPs, and PSNPs. Also may be set in SNPs.
- Old style and new style cannot be configured for the same scope (ISIS or interface)
- ***(IF) isis authentication send-only [{level-1 | level-2}]***
  Ignore authentications from peers, but send authenticated PDUs



IS-IS

Timers

Neighbor

(IF) isis hello-interval {<sec> | minimal} [level-1 | level-2]
Default hello is 10s for p2p and broadcast, and 3.3s for DIS on NBMA. For ***minimal*** Hello, the Holdtime is 1 sec

(IF) isis hello-multiplier <#> [{level-1 | level-2}]
Default multiplier is 3

(IF) isis lsp-interval <ms>
Time between consecutive LSPs. Default is 33ms

(ISIS) max-lsp-lifetime <sec> [{level-1 | level-2}]
Remaining Lifetime. Used to age out old LSPs. Lifetime is 1200sec. When lifetime expires, the LSP is purged from the network

(ISIS) lsp-refresh-interval <sec> [{level-1 | level-2}]
LSP Refresh. Specifies the time (default 15 min) a router will wait before refreshing its own LSP

(IF) isis retransmit-interval <sec>
Interval between retransmissions of the same LSP if ACK is not received (only p2p, no effect on LAN). Default is 5s. The newer LSP is flooded periodically until the neighbor acknowledges by sending PSNP or by sending an LSP that is the same or newer than the LSP being flooded.

(IF) isis retransmit-throttle-interval <msec>
Delay between retransmitted LSPs on p2p link. Default is 33ms

Flooding

Routers know how to reach system IDs within an area. Between areas, routers know how to reach the backbone, and the backbone knows how to reach other areas

Link State PDU
- Describe the router with all directly connected networks. One set per router and one set per each LAN network
- An IS can generate up to 256 LSPs (fragments) at each level numbered from 0 to 255
- LPS 0 has special properties, including (ATT bit)

Sequence Number PDU (SNP) contains a summary description of one or more LSPs

Complete SNP
- Used to periodically describe the LSPDB over LAN and only initially for p2p
- *(IF) isis csnp-interval <sec>*
  DIS multicasts CSNPs every 10 seconds. No ACK on broadcast

Partial SNP
- ACK for CSNPs on p2p links. No ACK on LAN
- Contains LSPs requested by the neighbor on LAN

On multiaccess networks CNSPs sent periodicaly by DIS are checked by each IS. If the IS has more recent version of LSP it is flooded. If older version is in local LSPDB then PSNP is sent to request updated LSP from DIS

(ISIS) set-overload-bit [on-startup <sec> [wait-for-bgp]] [suppress {external | interlevel}]
Clear OL bit after defined time since the router starts or once BGP converges

(ISIS) ispf [level-1 | level-2 | level-1-2] [<sec>]
Incremental SPF allows the system to recompute only the affected part of the tree. Seconds define after that time since configuring ISPF this feature is activated (default 120 sec)

(ISIS) fast-flood <number of LSPs>
Flood number of LSPs before starting SPF computation. The router should always flood (at least) the LSP that triggered SPF before the router runs the SPF computation

(ISIS) ip fast-convergence
Flood first 5 LSPs before starting SPF computations

(isis) partition avoidance <area-tag>
Router withdraws L1 prefix from L2 area when it no longer has any active adjacencies to that L1 area

(ISIS) ip route priority high tag <value>
Priority-Driven IP Prefix RIB Installation. Assigns a high priority to prefixes associated with the specified tag value. High-priority prefixes (loopbacks) are updated first in RIB. Medium priority - any /32 prefixes which is not a priority prefix. Low priority - all other prefixes

show isis spf-log

show isis database [{level-1 | level-2}]

show isis database <LSP ID> detail

```
R5#show isis database
IS-IS Level-1 Link State Database:
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
R3.00-00              0x00000031   0xC23C        702  Attached bit 1/0/0
R4.00-00              0x0000003A   0x6467        703               0/0/0
R5.00-00  This router * 0x00000022 0xD470        1174              0/0/0
```

Routing

(IF) isis bfd [disable]

Inter-level routing goes via the RIB. If it is not in the routing table, it is not advertised from L1 to L2

Internal routes are to destinations within an ISIS domain (L1 and L2). External routes are to destinations outside of an ISIS domain (redistributed)

(IF) isis tag <tag>
Sets a tag for IP subnets configured under this interface (ISIS has to be enabled on that interface). Tag – 4 bytes, carried in sub-TLV 1 of TLV 135

Explicit redistribution between IS-IS instances is prohibited

(ISIS) redistribute static ip ...
If the ***ip*** keyword is not used, then CLNS networks are redistributed. Default type is L1 and Internal

Leaking

(ISIS) redistribute isis ip level-2 into level-1 distribute-list <100-199>
Route leaking is possible, routes from L2 installed in L1 area (ia – inter-area)

The up/down bit (in TLV 128, 130, and 135) is used to indicate if the route has been leaked. It prevents routing loops. An L1/L2 router does not re-advertise into L2 any L1 routes that have the up/down bit set

(ISIS) redistribute maximum-prefix <max> [<%>] [warning-only | withdraw]
75% is a default threshold. If withdraw is used, all redistributed prefixes are removed from ISIS database when threshold is reached

(ISIS) lsp-full suppress {[external] [interlevel] | none}
Controls which routes are suppressed when the link-state PDU becomes full

(ISIS) summary-address <net> <mask> [{level-1 | level-2}] [metric <#>]
Internal route summarization is possible only at L1 => L2. External summarization is possible everywhere, during redistribution. Summarization must be configured the same on all L1/L2 routers. More specific routes are supressed. The metric is taken from the smallest metric

(IF) no isis advertise-prefix
ISIS can be enabled on interface, but the prefix of that interface will not be advertised

(isis) advertise-passive-only
Large-scale solution for fast-convergence by limiting routes advertised. Exclude IP prefixes of connected networks in LSP advertisements.

show isis rib [<prefix>]

Default route

(ISIS) set-attached-bit route-map <name>
Bu default L2 router sets the ATT (attached bit) in L1 LSPs (ONLY IF IT HAS NEIGHBORS IN OTHER AREAS) to define an area boundry (L1 installs 0/0 to the router with shortest metric). The bit can be set conditionally if specific CLNS routes are present in CLNS table

(ISIS) default-information originate [route-map <name>]
By default 0/0 is advertised only with L2 LSPs. The default does not have to be in routing table

When routes are redistributed into ISIS domain, the default route is not automaticaly redistributed.

(RM) set level level-1
Advertise 0/0 to L1 routers. Watch for L1L2 links, as L1 is more preferred than L2, you can accidentaly override old 0/0. Do it on the router which has L2-only and L1L2 interface, not L1L2 and L1 interfaces. 0/0 has better preference than LSP with ATT bit

Fragmented LSP
R5.00-00

BGP

Features

- BGP does not have it own transport (protocol number). It's a reachability application, which relies on IGP
- BGP is a TCP-based application, so it can be optimized with MTU, MSS, Windows Size, Selective Ack, etc.
- TCP/179 destination, random local port, path-vector protocol
- AD for eBGP is 20, AD for iBGP is 200, AD for backdoor routes is 200
- ***(BGP) distance bgp <ext> <int> <local and backdoor>***
  Set distance for all prefixes
- ***(BGP) distance <AD> <source IP> <source mask> [<acl>]***
  Set distance for specific prefixes (ACL) received from specific peer
- BGP has own internal queue 100 packets. It cannot be changed. It is not the same queue as ***hold-queue <x> in***
- ***(G) router bgp <as#>***
  AS can be either plain integer (32bit) or x.y notation. By default AS will be shown in config as integer, reladless of notation used
- 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

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** – session is successfuly established

Timers

- **(BGP) bgp scan-time <sec>**
  BGP scanner (verifying NH reachability) interval, default 60 sec
- **(BGP) neighbor <ip> advertisement-interval <sec>**
  If updates are ready to be sent to peers, they are delayed until advertisement interval ends. Default 5 sec – iBGP, 30 sec - eBGP
- ***(BGP) timers bgp <keepalive> <hold> [<min-hold>]***
  (BGP) neighbor <ip> timers <keepalive> <hold> [<min-hold>]
  By default lower negotiated holdtime is used. To prevent low holdtimes set by neighbor, minimum accepted can be defined. Keepalive every 60 sec, Holdtime 180 sec. Changing timers requires session restart (***clear ip bgp <neighbor>***)for changes to be applied

Header

- Marker: all 1s if no Auth
- Message Types: OPEN (1), UPDATE (2), NOTIFICATION (3), KEEPALIVE (4), ROUTE-REFRESH (5)
- Empty header is a keepalive

Marker (16 B)	
Length (16)	Type (8)

Version (8)			
My ASN (16)	Hold Down (16)		
BGP Identifier (32)			
Opts Len (8)	Opt Type (8)	Opt Len (8)	Opt Val (...)

Unfeasable routes length (32)
Withdrawn routes (var)
Total path attribute length (32)
Path attributes (var)
NLRI (var)

OPEN

- Optional parameters are formated as TLVs (type, length, value)
- Capabilities are advertised in OPEN message (Code, Length, Value)

UPDATE

- A value of 0 for unfeasable routes length indicates that no routes are being withdrawn, and that the withdrawn routes field is not present in this UPDATE message
- Withdrawn routes is a list of prefixes to be withdrawn
- A value of 0 for Total Path Attribute Length indicates that NLRI field is not present in UPDATE
- NLRI lentgth is not explicitly defined but can be calculated as: UPDATE Length - 32 - Total Path Attributes Length - Unfeasible Routes Length
- The min. length of UPDATE message is 23B: 19B fixed header + 2B for the Unfeasible Routes Length + 2B for the Total Path Attribute Length (when the value of Unfeasible Routes Length is 0 and the value of Total Path Attribute Length is 0)
- All path attributes contained in UPDATE messages apply to destinations carried in the NLRI field
- Path attributes is a list of TLVs.

Decision Process

1. **Largest Weight** (localy originated paths: 32768, other 0)
2. **Largest Local-Preefernce** (default 100)
3. **Prefer local paths** (preference order: *default-originate* in neighbor, *default-information-originate* in global, *network, redistribute, aggrgegate*)

▲ *Largest*

▼ *Smallest*

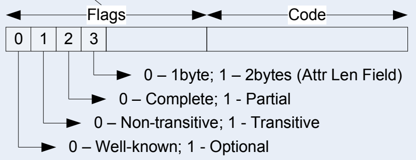
4. **Shortest AS_PATH** (unless *bgp bestpath as-path ignore*; AS_SET is 1; AS_CONFED_SEQUENCE and AS_CONFED_SET are not counted)
5. **Lowest origin code** (0-IGP, 1-EGP, 2-Incomplete)
6. **Lowest MED** (*bgp always-compare-med*; *bgp bestpath med-confed*; *bgp bestpath med missing-as-worst*; *bgp deterministic-med*) default 0
7. **eBGP prefered over iBGP** (Confederation paths are treated as internal paths)
8. **IGP metric to Next-Hop** (lowest cost unless *bgp bestpath igp-metric ignore*)
9. **Multipathing** (*bgp bestpath as-path multipath-relax* – allow different AS paths to form multipath, best path is still advertised)

Tie-breakers

10. **Oldest external path** (flap prevention). Skipped if *bgp bestpath compare-routerid*
11. **Lowest Router-ID** (unless *no bgp bestpath compare-routerid*)
12. **Shortest Cluster-List** (RR environment)
13. **Lowest neighbor address**

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	O NT
12	Advertiser	
13	RCID_Path/Cluster_Id	
14	MP-reachable NLRI	O NT
15	MP-unreachable NLRI	O NT
16	Extended Communities	
17	AS4_PATH	O T
18	AS4_AGGREGATOR	O T

WK – well-known; M – mandatory; D - discretionary
O – optional; T – transitive; NT – non-transitive

BGP

Session

- If both routers start session at the same time, session initiated by router with higher RID stays, and the other one is dropped
- TLL is checked only during session establishment.
- ***(BGP) neighbor <ip> remote-as <as>***
  BGP packets are dropped if there is no neighbor defined locally
- ***(BGP) neighbor <ip> update-source <if>***
  By default outgoing interface's IP is used. The source must the same IP that the remote router uses as a ***neighbor*** (BGP does not see the topology, and it doesn't know all remote router's IPs). For iBGP use loopbacks
- ***(BGP) neighbor <ip> transport connection-mode {active | passive}***
  By default the router tries to establish session actively, and listens to incomming sessions
- ***(BGP) bgp update-delay <sec>***
  Upon establishing session and exchanging OPEN message router starts Read-only mode during which it does not perform best-path selection. The reason is to wait until neighbor sends all prefixes. Default 30 sec

eBGP

- TTL is 1. Peers must be directly connected
- If remote AS is different than ours, the session is eBGP
- Router sending an update sets NH to own outgoing IP
- ***(BGP) neighbor <ip> disable-connected-check***
  TTL stays 1. Used for directly connected multihop eBGP peers with loopback-based session
- ***(BGP) neighbor <ip> ebgp-multihop [<ttl>]***
  TTL in IP packet changed from 1 to a defined value. There must be a specific route to remote peer. Default route will not work
- Load-balancing
  - Next-hop router for each multipath must be different
  - All attributes of redundant paths must be the same
  - ***(BGP) maximum-paths [ibgp] <up-to-6>***
    By default eBGP does not perform load balancing. Only one path is installed in routing table. Multipath applies only to eBGP and external confederation peer
  - ***(BGP) bgp additional-paths install***
    Enable backup path to be stored in table use. Multi-path must be disabled, as BGP will install both paths if they are equal. ***show ip bgp repair-paths <prefix>***

iBGP

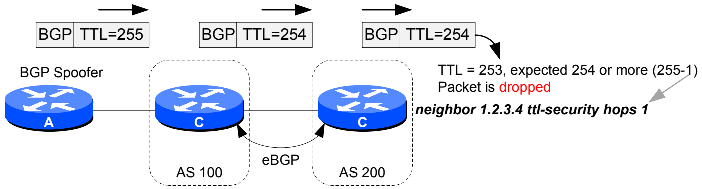
- TTL is 255. Peers do not have to be directly connected, IGP provides remote IP reachability
- If remote AS is the same as ours, the session is iBGP
- Routes received from other iBGP peer are not sent to iBGP peers
- Next-Hop is not modified when route is passed withing iBGP domain (in RR too, we do not want RR to be on the path, we want shortest path to exit point)

Automatic neighbors

- ***(BGP) bgp listen limit <#>***
  Limit number of automatic neighbors
- ***(BGP) bgp listen range <prefix> peer-group <name>***
  Prefix defines from which addresses session is accepted
- ***(BGP) neighbor <group-name> alternate-as <list of ASes)***
  Accept neighbor in defined ASes only (list separated with space)

Security

- MD5 Auth
  - ***(BGP) neighbor <ip> password <string>***
    MD5 authentication is applied on the TCP psuedo-IP header, TCP header and data
  - TCP uses SN and ACK numbers, along with the BGP neighbor password to create a 128 bit MD5 hash, which is included in the packet in a TCP header option 19 field
  - When BGP session with MD5 travels through a firewall, you must disable TCP random sequence number feature on FW (usualy enabled by default). It changes the TCP sequence number of the incoming packets before it forwards them. Then checksums for MD5 do not match
- TTL check
  - Both sides must have this feature configured
  - Does not prevent attacks from the same segment or distance
  - **(BGP) neighbor <ip> ttl-security hops <#>**
    Reverse TTL logic. BGP will establish session only if TTL in IP header is equal to or greater than (TTL – hop) value configured for session. This command defines number of hops that are between peers. If TTL 255 is expected, <hop> should be 1 (checked after local router decrements TTL)
  - Protects only incoming packets. Supported only for eBGP. If multihop session is to be protected, ebgp-multihop must be disabled (mutually exclusive)



TTL = 253, expected 254 or more (255-1)
Packet is dropped

neighbor 1.2.3.4 ttl-security hops 1

Fast Session Deactivation

- Can also track peers' IPs, not only next-hops. Peer's IP can be tracked only if host route is present. If peer's IP is aggregated, this feature will not work.
- ***(BGP) bgp fast-external-fallover***
  Fast External Fallover Enabled by default. If turned off, does not react to connected interface going down, waits for holdtime to expire. Only for p2p connections
- ***(BGP) neighbor <ip> fall-over [bfd] [route-map <name>]***
  Event-driven, per neighbor. If we lose our **/32** route to the peer (multihop eBGP), tear down the session. No need to wait for the hold timer to expire. Similiar to fast external fallover for p2p sessions. Route-map can define prefixes (prefix-list) which must exist in a routing table, pointing to the peer (/32 by default), otherwise session is torn down
- Should be enabled on both sides, otherwise one side reacts fast, but the other waits for a deadtime

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

MTU

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

BGP

Confederation

As a loop prevention, AS_CONFED_SEQUENCE and AS_CONFED_SET is introduced. Each AS adds own sub-AS to path. {65001 65002}. Confed AS-set is counted as 1 AS in the path

When an 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

NEXT_HOP, MED, LOCAL_PREF are left untouched between sub-ASes. Common IGP is recommended

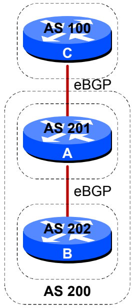
Full-mesh rule applies inside sub-as. RR can be used inside sub-AS to limit iBGP sessions

The session between Sub-ASes is an eBGP session with all eBGP rules applied

Route preference: ext eBGP -> confed ext eBGP -> iBGP

router bgp <id> (private AS)
bgp confederation identifier <id> (real AS)
bgp confederation peers <as> <as> (sub-ASes)

- Real AS is used for eBGP sessions
- Sub-ASes are all other ASes exluding local
- Peers configured only on Sub-AS eBGP routers



```
R2#sh ip bgp 55.55.55.0
BGP routing table entry for 55.55.55.0/24, version 11
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Confed Path, 120 is our neighbor
  (120 110) 70000                             Peer IP    Peer RID
    4.4.4.4 (metric 131072) from 3.3.3.3 (3.0.0.0)
  NH  Origin IGP, metric 0, localpref 100, valid, confed-external, best
      rx pathid: 0, tx pathid: 0x0
```

Route Reflector

Route replectors are mainly used to limit full-mesh sessions for iBGP, but it hides the topology (paths)

RR should be redundant. One cluster or many clusters depends on the design and requirements

RR advertises only the best path. In case of primary path failure, the convergence is slow. Also, underterministic path may be introduced, as some routers will not leard alternate paths

CLLUSTER_LIST updated by RR with CLUSTER_ID (usualy router ID) when RR sends route to a client. Loop avoidance, RR drops update with own Cluster ID

ORIGINATOR_ID (client's router ID) added by RR for updates sourced by a client. RR will not send update to the same peer as originator-id. A router will drop an update with own originator-id set in received update (from another client or RR)

RR can be implemented hierarchicaly. RR can be another RR's client

Physical path should follow RR-to-Client path to avoid blackholing and loops

Update from non-client is reflected to clients and eBGP peers

Update from eBGP is reflected to clients and non-clients

Update from client is reflected to non-clients, clients and eBGP peers

Route-reflector in different cluster is a non-client for local route-reflector

(BGP) neighbor <ip> route-reflector-client
Define a client on RR. Client is not aware of being a client, no additional configuration required

(BGP) bgp cluster-id <id>
If not set, it is a router ID. Set to the same ID if there are more than one RRs in a cluster

Connections between clusters must be made between the route reflectors, not between clients, because clients do not examine the CLUSTER_LIST (loop prevention)

(BGP) no bgp client-to-client reflection
Should be configured when clients are fully meshed

```
R1#show ip bgp 55.55.55.0
BGP routing table entry for 55.55.55.0/24, version 5
Paths: (1 available, best #1, table default)  Table version
  Advertised to update-groups:
     2
 Path esh Epoch 2              We are the RR
  70000, (Received from a RR-client)
    4.4.4.4 (metric 130816) from 4.4.4.4 (4.4.4.4)
 NH   Origin IGP, metric 0, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
```

```
R2#show ip bgp 55.55.55.0
BGP routing table entry for 55.55.55.0/24, version 18
Paths: (1 available, best #1, table default)  Table version
  Not advertised to any peer
  Refresh Epoch 1
  70000
    4.4.4.4 (metric 131072) from 1.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 4.4.4.4, Cluster list: 1.1.1.1
      rx path Router originating id: 0x0        RR cluster-ID
              the update
```

Peer-group

(BGP) neighbor <name> peer-group
Define peer-group. Common parameters can be defined per group

(BGP) neighbor <ip> peer-group <name>
Assign peer to a 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

After policy change is applied, update groups are automatically recalculated after 3 min (if mistake is made, it can be rolled back). Or, manual refresh can be done using ***clear ip bgp <ip> soft out***

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

(BGP) neighbor <ip> inherit peer-session <name>
One directly inherited template per peer

(BGP) template peer-session <name>

(TMPL) 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)

show ip bgp template peer-session

Peer policy

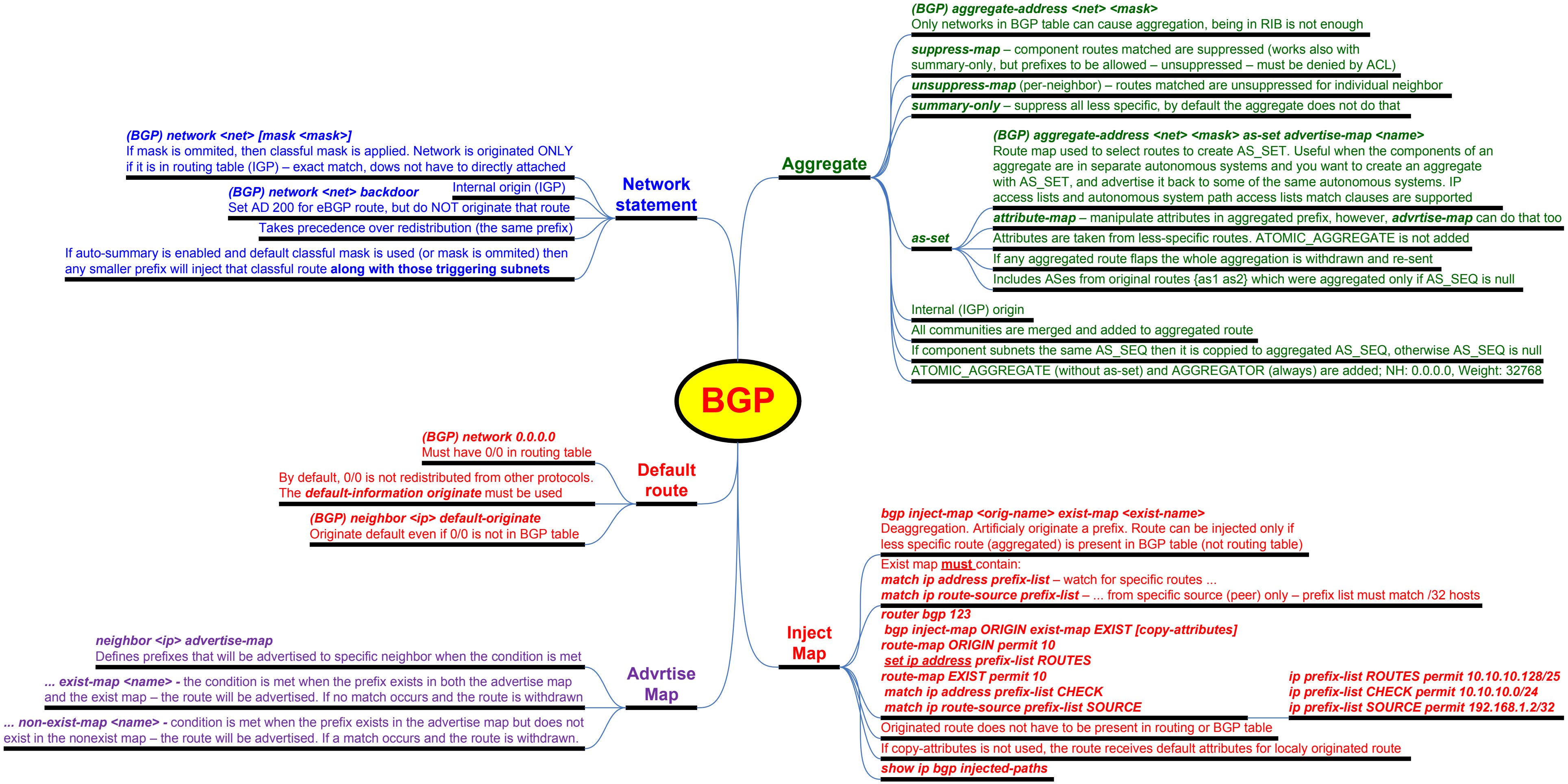
Up to 8 policy templates daisy-chain inherited

Inheritance is sequenced (starts with lowest) – ALL ENTRIES ARE EXECUTED

(TMPL) inherit peer-policy <name> <seq>

(BGP) neighbor <ip> inherit peer-policy <name>

show ip bgp template peer-policy



BGP

Network statement

- *(BGP) network <net> [mask <mask>]*
  If mask is ommited, then classful mask is applied. Network is originated ONLY if it is in routing table (IGP) – exact match, dows not have to directly attached
- Internal origin (IGP)
- *(BGP) 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 (or mask is ommited) then any smaller prefix will inject that classful route **along with those triggering subnets**

Aggregate

- ***(BGP) aggregate-address <net> <mask>***
  Only networks in BGP table can cause aggregation, being in RIB is not enough
- ***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, by default the aggregate does not do that
- ***as-set***
  - ***(BGP) aggregate-address <net> <mask> as-set advertise-map <name>***
    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, however, ***advrtise-map*** can do that too
  - Attributes are taken from less-specific routes. ATOMIC_AGGREGATE is not added
  - If any aggregated route flaps the whole aggregation is withdrawn and re-sent
  - Includes ASes from original routes {as1 as2} which were aggregated only if AS_SEQ is null
- Internal (IGP) origin
- All communities are merged and added to aggregated route
- If component subnets the same AS_SEQ then it is coppied to aggregated AS_SEQ, otherwise AS_SEQ is null
- ATOMIC_AGGREGATE (without as-set) and AGGREGATOR (always) are added; NH: 0.0.0.0, Weight: 32768

Default route

- ***(BGP) network 0.0.0.0***
  Must have 0/0 in routing table
- By default, 0/0 is not redistributed from other protocols. The ***default-information originate*** must be used
- ***(BGP) neighbor <ip> default-originate***
  Originate default even if 0/0 is not in BGP table

Advrtise Map

- ***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.

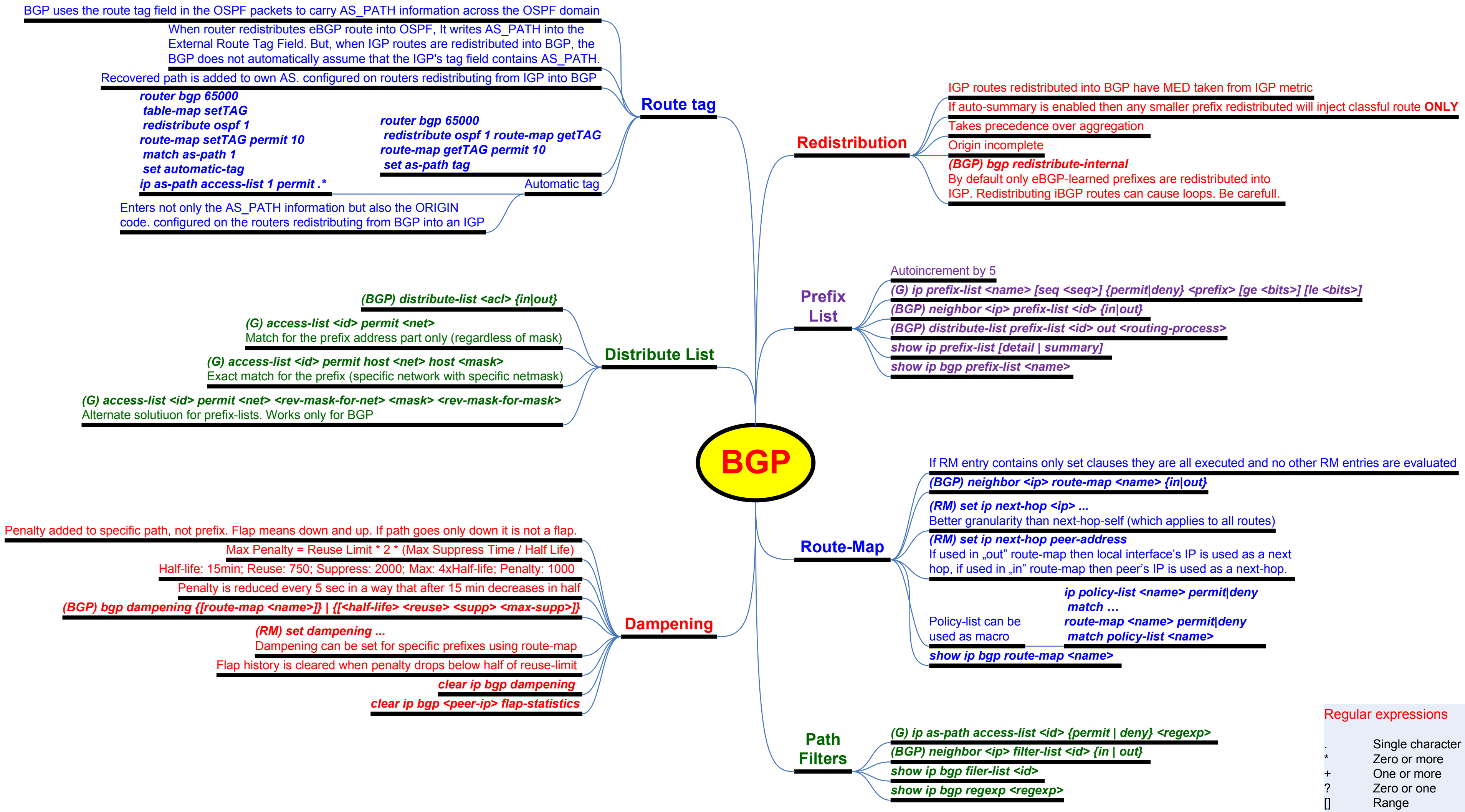
Inject Map

- ***bgp inject-map <orig-name> exist-map <exist-name>***
  Deaggregation. Artificialy originate a prefix. Route can be injected only if less specific route (aggregated) is present in BGP table (not routing table)
- Exist map **must** contain:
  match ip address prefix-list – watch for specific routes ...
  match ip route-source prefix-list – ... from specific source (peer) only – prefix list must match /32 hosts
- ***router bgp 123***
  bgp inject-map ORIGIN exist-map EXIST [copy-attributes]
  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 10.10.10.128/25***
    ip prefix-list CHECK permit 10.10.10.0/24
    ip prefix-list SOURCE permit 192.168.1.2/32
- Originated route does not have to be present in routing or BGP table
- If copy-attributes is not used, the route receives default attributes for localy originated route
- ***show ip bgp injected-paths***

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



BGP

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
 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 .*
```

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

Automatic tag

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

Redistribution

IGP routes redistributed into BGP have MED taken from IGP metric

If auto-summary is enabled then any smaller prefix redistributed will inject classful route **ONLY**

Takes precedence over aggregation

Origin incomplete

(BGP) bgp redistribute-internal
By default only eBGP-learned prefixes are redistributed into IGP. Redistributing iBGP routes can cause loops. Be carefull.

Prefix List

Autoincrement by 5

(G) ip prefix-list <name> [seq <seq>] {permit|deny} <prefix> [ge <bits>] [le <bits>]

(BGP) neighbor <ip> prefix-list <id> {in|out}

(BGP) distribute-list prefix-list <id> out <routing-process>

show ip prefix-list [detail | summary]

show ip bgp prefix-list <name>

Distribute List

(BGP) distribute-list <acl> {in|out}

(G) access-list <id> permit <net>
Match for the prefix address part only (regardless of mask)

(G) access-list <id> permit host <net> host <mask>
Exact match for the prefix (specific network with specific netmask)

(G) access-list <id> permit <net> <rev-mask-for-net> <mask> <rev-mask-for-mask>
Alternate solutiuon for prefix-lists. Works only for BGP

Route-Map

If RM entry contains only set clauses they are all executed and no other RM entries are evaluated

(BGP) neighbor <ip> route-map <name> {in|out}

(RM) set ip next-hop <ip> ...
Better granularity than next-hop-self (which applies to all routes)

(RM) 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.

Policy-list can be used as macro

```
ip policy-list <name> permit|deny
 match ...
route-map <name> permit|deny
 match policy-list <name>
```

show ip bgp route-map <name>

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 decreases in half

(BGP) bgp dampening {[route-map <name>]} | {[<half-life> <reuse> <supp> <max-supp>]}

(RM) set dampening ...
Dampening can be set for specific prefixes using 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

Path Filters

(G) ip as-path access-list <id> {permit | deny} <regexp>

(BGP) neighbor <ip> filter-list <id> {in | out}

show ip bgp filer-list <id>

show ip bgp regexp <regexp>

.*	Anything
^$	Empty path (local AS)
^111_	Learned from AS 111
_111$	Originated in AS 111
_111_	Transited AS
^[0-9]+$	Any directly connected AS

Regular expressions

.	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

BGP

BGP Table

Table version changes when prefix is received/withdrawn, and best path algorithm is run, new paths appear, and routes are installed in RIB table (change in paths)

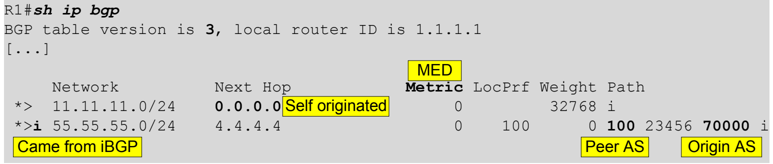
(BGP) bgp suppress-inactive
By default disabled, so inactive routes (not installed in RIB via BGP) are advertised

(BGP) bgp advertise-best-external
If external route is the best, and local BGP has alternate path, means local router is also an exit point, so advertise second best extenral route anyway. Used in RR environment, when RR select another bets path and advertises to local router. Does NOT work with PIC. Routes marked with „x”

(BGP) neighbor <ip> maximum-prefix <#> [<thrhld %>] [warning-only] [restart <sec>]
Limit number of prefixes per-neighbor

show ip bgp neighbor <ip> {routes | advertised-routes | received-routes}
Routes sent to the peer, received and installed, and received and not processed (requires soft-reconfig)

show ip bgp rib-failure
Route is in routing table, but not installed as BGP, however received via BGP



```
R1#sh ip bgp
BGP table version is 3, local router ID is 1.1.1.1
[...]

     Network          Next Hop            Metric LocPrf Weight Path
 *>  11.11.11.0/24    0.0.0.0                  0         32768 i
 *>i 55.55.55.0/24    4.4.4.4                  0    100      0 100 23456 70000 i
```

Prefix refresh

Soft Reconfig

All received peer’s prefixes are stored in local table (marked as received-only). When policy is changed, they do not have to be re-sent. Requires additional memory

(BGP) neighbor <ip> soft-reconfigation inbound

clear ip bgp {<id> | *} soft {in|out}

Route Refresh

Replacement for soft-reconfiguration. Negotiated with OPEN message

clear ip bgp {<id> | *} {in | out}
Dynamicaly request Adj-RIP-out from peer for specific AFI/SAFI

ORF

Outbound Route Filtering. Only for individual peers. Negotiated in OPEN message

Requires prefix-list configuration (the only method supported)

BGP speaker can install the inbound prefix list filter on the remote peer’s control plane as an outbound filter. No need to send all routes to the peer for him to do filtering (but must process all unneeded prefixes, and waste CPU)

(BGP) neighbor <ip> capability orf prefix-list {send | receive | both}
Send means the request (filter) is sent from the customer to ISP, which receives it

(BGP) 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

PIC

Prefix Independent Convergence speeds up convergence by finding a second best path. It is recommended to set repair paths for important prefixes, not all in global routing table

PIC makes sense if BFD is used for fast failure detection, otherwide regular update will refresh routes

(BGP) neighbor <ip> advertise diverse-path [backup] [mpath]

(BGP) bgp bestpath igp-metric ignore
Use on RR, so it advertises more than one best path

(BGP) bgp additional-paths [send] [receive]

(BGP) bgp additional-paths install
Install paths, selected by the ***select*** command, into the RIB and CEF. Can be per-AF

(BGP) bgp additional-paths select {best-external | backup | best <#> | all}
Calculate second best paths. Paths can be limited in case of small memory and TCAM resources

show ip cef <prefix> detail will show backup paths

Backup paths are marked with „*>bi” (backup/repair path) in ***show ip bgp <prefix>***

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

After End of RIB marker (empty withdrawn NLRI TLV) is received, best-path algorithm is run, and routing table is updated. Stale information is removed from FIB

(BGP) bgp graceful-restart
Enable graceful restart capability globally for all BGP neighbors

(BGP) neighbor <ip> ha-mode graceful-restart
Enable graceful restart capability per neighbor

(BGP) bgp graceful-restart restart-time <sec>
Maximum time (120 sec default) router will wait for peer to return to normal operation

(BGP) bgp graceful-restart stalepath-time <sec>
Maximum time (360 sec default) router will hold stale paths for a restarting peer

IPv6

(BGP) address-family ipv6 unicast
AFI 2, SAFI 1

Two new optional, non-transitive attributes: Multiprotocol Reachable NLRI (MP_REACH_NLRI) – Type Code 14; Multiprotocol Unreachable NLRI (MP_UNREACH_NLRI) – Type Code 15

The transport can be IPv4 or IPv6, both transports can exchange both NLRIs

In pure IPv6 environment router-id must be set manually

router bgp 10
 neighbor 2002:10::1 remote-as 20 <= activate TCP session
 address-family ipv6 unicast
  neighbor 2002:1::1 activate <= activate AFI 2 / SAFI 1

IPv4 control transport

router bgp 10
 neighbor 10.0.0.1 remote-as 20
 address-family ipv6 unicast
  neighbor 10.0.0.1 activate <= IPv4 control transport

By default NH is set to IPv4 encoded IP ::FFFF:10.0.0.1 (non-existent in FIB, so route is not installed in routing table)

NH can be set with an inbound route-map to a connected address

(BGP) no bgp default ipv6-nexthop
Must be set on advertising router, then NH is set to a connected address (global prefered over link-local)

So, IPv4 transport (TCP session) still requires IPv6 link addresses

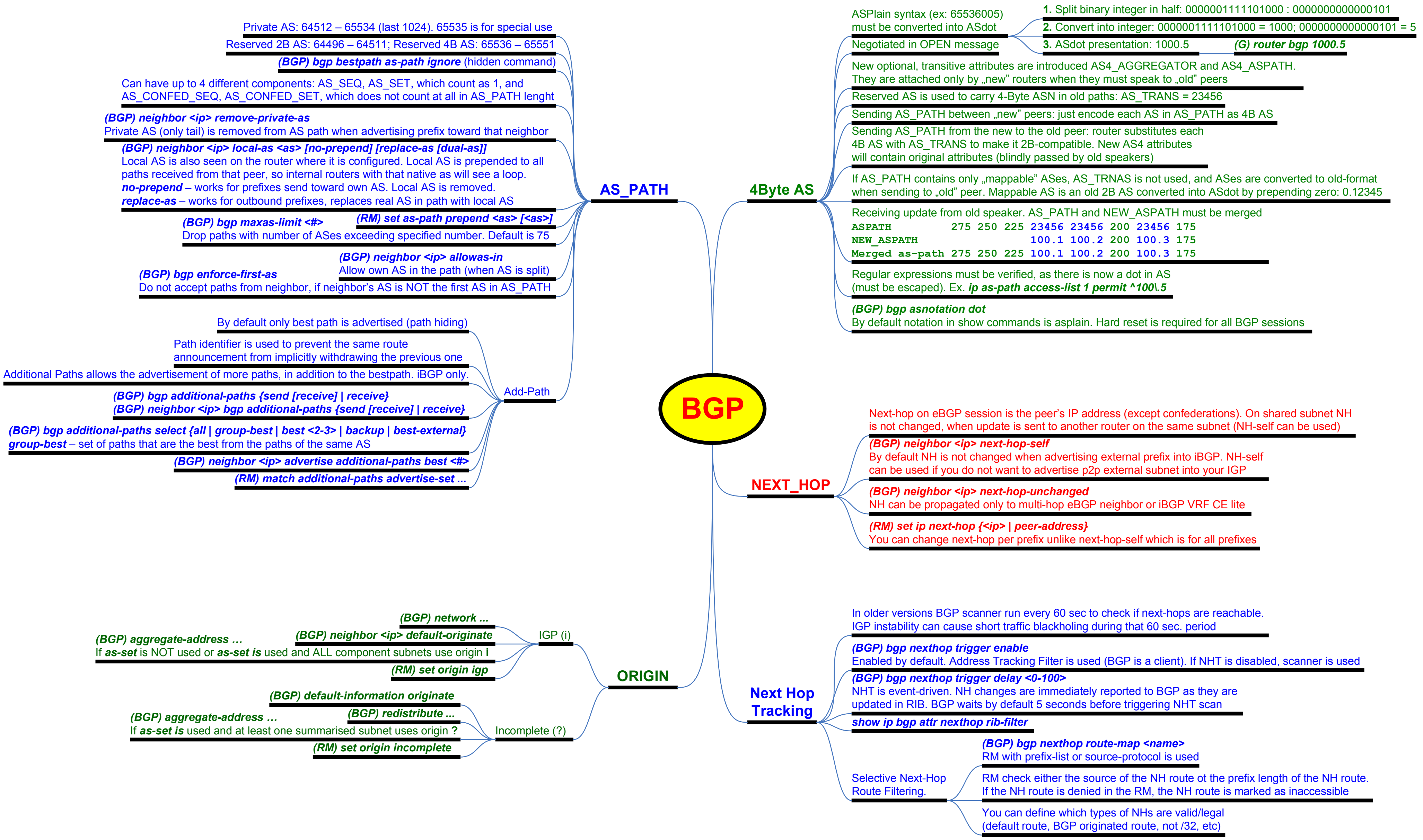
(BGP) neighbor FE80::1%GigabitEthernet0/0 remote-as 20
Neighbor must be global address, not link-local, as interface cannot be identified. To establish the session using link-local addresses use % notation

The next-hop field contains a global IPv6 address and potentially a link-local IPv6 address (directly connected session)

Next hop in BGP table is the neighbor (also link-local address if session is established on link-local), but in routing table it is always a link-local

When only a link-local next-hop address is present, this needs to be changed to a global address for the iBGP update

show bgp ipv6 unicast summary



BGP

AS_PATH

Private AS: 64512 – 65534 (last 1024). 65535 is for special use

Reserved 2B AS: 64496 – 64511; Reserved 4B AS: 65536 – 65551

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

Can have up to 4 different components: AS_SEQ, AS_SET, which count as 1, and
AS_CONFED_SEQ, AS_CONFED_SET, which does not count at all in AS_PATH lenght

(BGP) neighbor <ip> remove-private-as
Private AS (only tail) is removed from AS path when advertising prefix toward that neighbor

(BGP) 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) bgp maxas-limit <#>
Drop paths with number of ASes exceeding specified number. Default is 75

(BGP) neighbor <ip> allowas-in
Allow own AS in the path (when AS is split)

(BGP) bgp enforce-first-as
Do not accept paths from neighbor, if neighbor's AS is NOT the first AS in AS_PATH

Add-Path

By default only best path is advertised (path hiding)

Path identifier is used to prevent the same route
announcement from implicitly withdrawing the previous one

Additional Paths allows the advertisement of more paths, in addition to the bestpath. iBGP only.

(BGP) bgp additional-paths {send [receive] | receive}
(BGP) neighbor <ip> bgp additional-paths {send [receive] | receive}

(BGP) bgp additional-paths select {all | group-best | best <2-3> | backup | best-external}
group-best – set of paths that are the best from the paths of the same AS

(BGP) neighbor <ip> advertise additional-paths best <#>

(RM) match additional-paths advertise-set ...

4Byte AS

ASPlain syntax (ex: 65536005)
must be converted into ASdot

1. Split binary integer in half: 0000001111101000 : 0000000000000101

2. Convert into integer: 0000001111101000 = 1000; 0000000000000101 = 5

3. ASdot presentation: 1000.5 ***(G) router bgp 1000.5***

Negotiated in OPEN message

New optional, transitive attributes are introduced AS4_AGGREGATOR and AS4_ASPATH.
They are attached only by „new" routers when they must speak to „old" peers

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

Sending AS_PATH between „new" peers: just encode each AS in AS_PATH as 4B AS

Sending AS_PATH from the new to the old peer: router substitutes each
4B AS with AS_TRANS to make it 2B-compatible. New AS4 attributes
will contain original attributes (blindly passed by old speakers)

If AS_PATH contains only „mappable" ASes, AS_TRNAS is not used, and ASes are converted to old-format
when sending to „old" peer. Mappable AS is an old 2B AS converted into ASdot by prepending zero: 0.12345

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
```

Regular expressions must be verified, as there is now a dot in AS
(must be escaped). Ex. ***ip as-path access-list 1 permit ^100\.5***

(BGP) bgp asnotation dot
By default notation in show commands is asplain. Hard reset is required for all BGP sessions

NEXT_HOP

Next-hop on eBGP session is the peer's IP address (except confederations). On shared subnet NH
is not changed, when update is sent to another router on the same subnet (NH-self can be used)

(BGP) neighbor <ip> next-hop-self
By default NH is not changed when advertising external prefix into iBGP. NH-self
can be used if you do not want to advertise p2p external subnet into your IGP

(BGP) neighbor <ip> next-hop-unchanged
NH can be propagated only to multi-hop eBGP neighbor or iBGP VRF CE lite

(RM) set ip next-hop {<ip> | peer-address}
You can change next-hop per prefix unlike next-hop-self which is for all prefixes

ORIGIN

IGP (i)

(BGP) network ...

(BGP) neighbor <ip> default-originate

(BGP) aggregate-address ...
If ***as-set*** is NOT used or ***as-set is*** used and ALL component subnets use origin **i**

(RM) set origin igp

Incomplete (?)

(BGP) default-information originate

(BGP) redistribute ...

(BGP) aggregate-address ...
If ***as-set is*** used and at least one summarised subnet uses origin **?**

(RM) set origin incomplete

Next Hop Tracking

In older versions BGP scanner run every 60 sec to check if next-hops are reachable.
IGP instability can cause short traffic blackholing during that 60 sec. period

(BGP) bgp nexthop trigger enable
Enabled by default. Address Tracking Filter is used (BGP is a client). If NHT is disabled, scanner is used

(BGP) bgp nexthop trigger delay <0-100>
NHT is event-driven. NH changes are immediately reported to BGP as they are
updated in RIB. BGP waits by default 5 seconds before triggering NHT scan

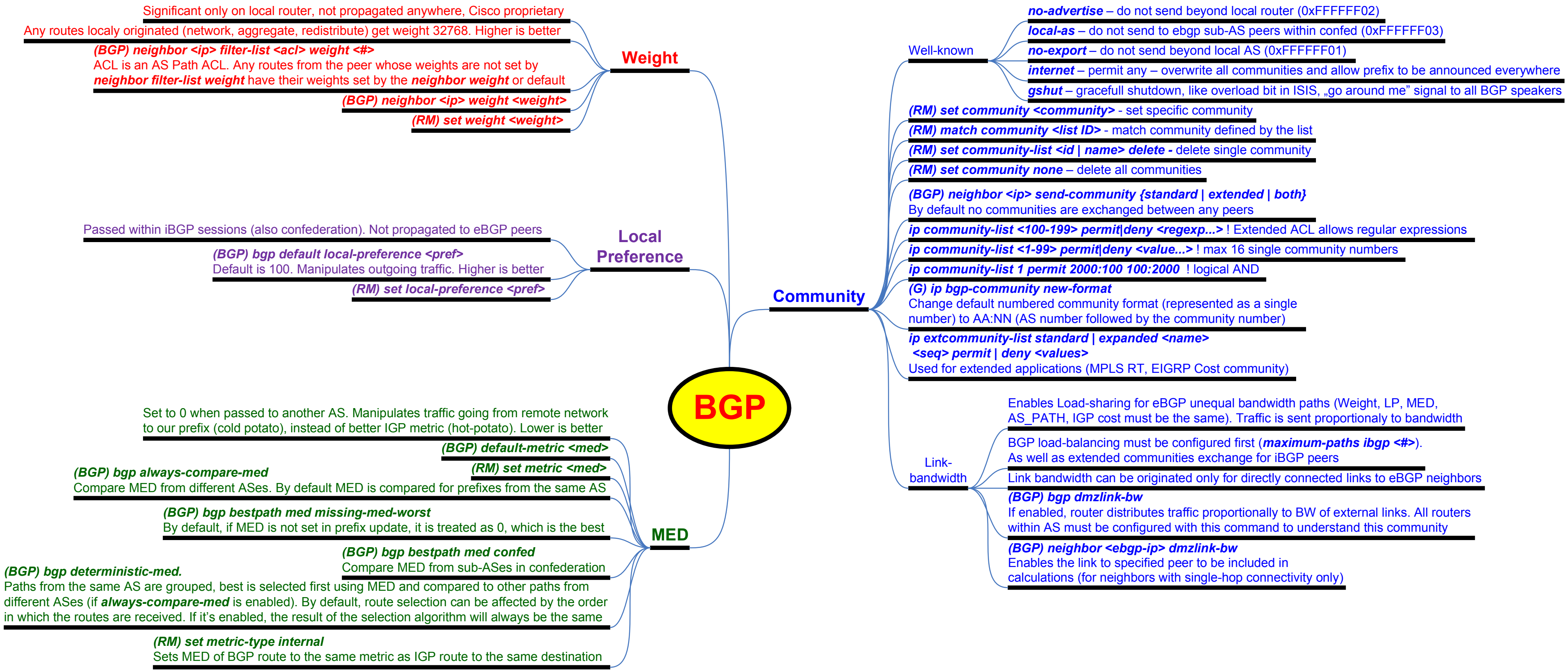
show ip bgp attr nexthop rib-filter

Selective Next-Hop
Route Filtering.

(BGP) bgp nexthop route-map <name>
RM with prefix-list or source-protocol is used

RM check either the source of the NH route ot the prefix length of the NH route.
If the NH route is denied in the RM, the NH route is marked as inaccessible

You can define which types of NHs are valid/legal
(default route, BGP originated route, not /32, etc)



BGP

Weight

Significant only on local router, not propagated anywhere, Cisco proprietary

Any routes localy originated (network, aggregate, redistribute) get weight 32768. Higher is better

(BGP) neighbor <ip> filter-list <acl> weight <#>
ACL is an AS Path ACL. Any routes from the peer whose weights are not set by ***neighbor filter-list weight*** have their weights set by the ***neighbor weight*** or default

(BGP) neighbor <ip> weight <weight>

(RM) set weight <weight>

Local Preference

Passed within iBGP sessions (also confederation). Not propagated to eBGP peers

(BGP) bgp default local-preference <pref>
Default is 100. Manipulates outgoing traffic. Higher is better

(RM) set local-preference <pref>

MED

Set to 0 when passed to another AS. Manipulates traffic going from remote network to our prefix (cold potato), instead of better IGP metric (hot-potato). Lower is better

(BGP) default-metric <med>

(RM) set metric <med>

(BGP) bgp always-compare-med
Compare MED from different ASes. By default MED is compared for prefixes from the same AS

(BGP) bgp bestpath med missing-med-worst
By default, if MED is not set in prefix update, it is treated as 0, which is the best

(BGP) bgp bestpath med confed
Compare MED from sub-ASes in confederation

(BGP) bgp deterministic-med.
Paths from the same AS are grouped, best is selected first using MED and compared to other paths from different ASes (if ***always-compare-med*** is enabled). By default, route selection can be affected by the order in which the routes are received. If it's enabled, the result of the selection algorithm will always be the same

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

Community

Well-known

- ***no-advertise*** – do not send beyond local router (0xFFFFFF02)
- ***local-as*** – do not send to ebgp sub-AS peers within confed (0xFFFFFF03)
- ***no-export*** – do not send beyond local AS (0xFFFFFF01)
- ***internet*** – permit any – overwrite all communities and allow prefix to be announced everywhere
- ***gshut*** – gracefull shutdown, like overload bit in ISIS, „go around me" signal to all BGP speakers

(RM) set community <community> - set specific community

(RM) match community <list ID> - match community defined by the list

(RM) set community-list <id | name> delete - delete single community

(RM) set community none – delete all communities

(BGP) neighbor <ip> send-community {standard | extended | both}
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

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

ip extcommunity-list standard | expanded <name>
<seq> permit | deny <values>
Used for extended applications (MPLS RT, EIGRP Cost community)

Link-bandwidth

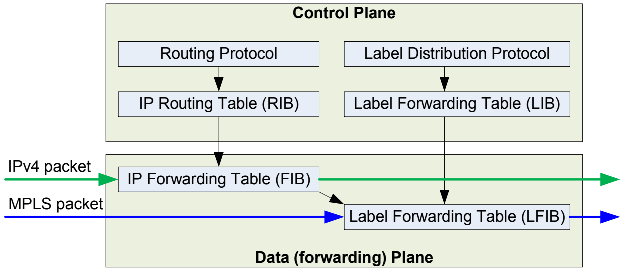
Enables Load-sharing for eBGP unequal bandwidth paths (Weight, LP, MED, AS_PATH, IGP cost must be the same). Traffic is sent proportionaly to bandwidth

BGP load-balancing must be configured first (***maximum-paths ibgp <#>***). As well as extended communities exchange for iBGP peers

Link bandwidth can be originated only for directly connected links to eBGP neighbors

(BGP) bgp dmzlink-bw
If enabled, router distributes traffic proportionally to BW of external links. All routers within AS must be configured with this command to understand this community

(BGP) neighbor <ebgp-ip> dmzlink-bw
Enables the link to specified peer to be included in calculations (for neighbors with single-hop connectivity only)


Control Plane
Routing Protocol
Label Distribution Protocol
IP Routing Table (RIB)
Label Forwarding Table (LIB)
IPv4 packet
MPLS packet
IP Forwarding Table (FIB)
Label Forwarding Table (LFIB)
Data (forwarding) Plane

MPLS

LRIB

- 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 LRIB 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
- ***show mpls ldp binding***

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}
- Binding can be created only if RIB (IGP advertisement) and LRIB (LDP advertisement) entries match. LSP endpoints must be /32, no summarization on the way

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, but it will not send larger packet to the interface (depending on the side with too low MTU, the „ICMP Fragmentation Needed and DF 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***

(IF) ip mtu 1500	1500	
(IF) mpls ip	1492	8
(IF) mpls mtu 1508	1500	8

Load balancing

- 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 mpls forwarding-table labels <label> exact-path ipv4 <src> <dst>***
  Displays which path the labeled patcked will take.

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.

Labels

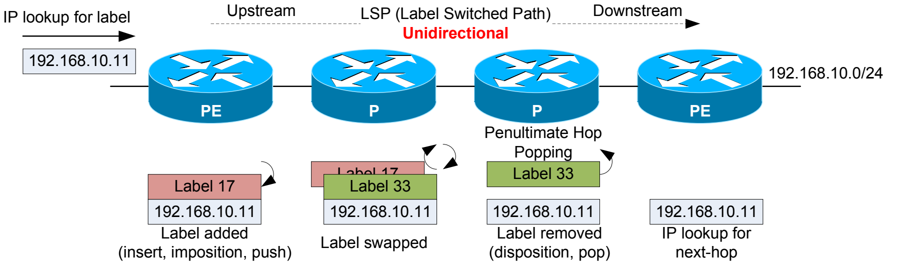
20	3	1	8
Label	**Exp**	**S**	**TTL**

◄—32 bits—►

Labels

- 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.
- MPLS can only use the label based on the route that is installed in routing table (igp next hop)
- 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
- 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
  - Eth 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
- 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)

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
  - 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

	L2 header		
Label stack	TE label	S=0	Top label
	LDP label	S=0	
	VPN label	S=1	Bottom label
	IP Header		
	Payload		

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 (discovery)

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 (TLV) is the same as the LDP Router-ID (LSR-ID). If multiple interfaces exist between LSRs, they all must use the same transport address, so it must be changed, or use loopback as a Router ID and the source (preferred to have dedicated loopbacks for MPLS, aside to regular loopbacks). Transport address must be reachable (via IGP)

Initialization messages (keepalive, distribution method, max PDU length, peer's LDP ID) are exchanged after TCP is established. Then keepalive messages every 60 sec. Labels are exchanged after first keeaplive received

(IF/G) mpls label protocol {tdp | ldp | both}
LDP is default. 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 interface

```
R1#show mpls ldp neighbor
    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 1.1.1.1:0
        TCP connection: 2.2.2.2.58085 - 1.1.1.1.646
        State: Oper; Msgs sent/rcvd: 28/29; Downstream
        Up time: 00:17:18
        LDP discovery sources:
          GigabitEthernet0/0, Src IP addr: 10.0.12.2
        Addresses bound to peer LDP Ident:
          10.0.12.2       2.2.2.2         10.0.23.2

R1#show mpls interfaces
Interface              IP          RSVP Tunnel    BGP Static Operational
GigabitEthernet0/0     Yes (ldp)        No        No  No     Yes
```

???

IPs assigned to interfaces (by default all)

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 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. Those tunnel endpoint MUST be /32 (loopbacks). Conditional propagation is not only for local prefixes but also for advertised by peers, so ACL must match appropriate range.

(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

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

(G) mpls label range <min> <max>
Default range is 16 – 100000. Use ***show mpls label range*** to verify. Reload may be required

show mpls ldp binding [advertised-acl]

show mpls ldp binding detail

Autoconfig

(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 (OSPF and ISIS), 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

If autoconfig is enabled for IGP, MPLS can be disabled globaly (***no mpls ip***) only if autoconfig is removed first

show mpls ldp neighbor password

LDP

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 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
- ***(G) mpls ldp logging password {configuration | rollover} [rate-limit <#>]***
  Display password configuration change or rollover events on LSR
- ***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.

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

IGP sync

- 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 and blackhole the traffic
- ***(OSPF) mpls ldp sync***
  Only OSPF supports synchronization (recommended best practice). 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 is not synced with LDP, but link is still announced with max cost (65536)
- ***show ip ospf mpls ldp interface <if>***
- ***show mpls ldp igp sync***

VRF

Features

- Customers' routes must be distinguished on PE routers. Virtual routing and forwarding (VRF) tables are used
- ***(G) vrf definition <name>***
  New format, supports IPv4 and IPv6
- ***(G) ip vrf <name>***
  Old format, IPv4 only
  - ***(G) vrf upgrade-cli multi-af-mode common-policies***
    Change ***ip vrf*** into ***vrf definition*** configuration
- ***(IF) ip vrf forwarding <VRF name>***
  Assign VRF to interface. Only IPv4 will be REMOVED if ***ip vrf*** was used to create the VRF. If ***vrf definition*** was used, both addresses are removed (depending on address family configured inside VRF). 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***

VRF Lite

- Only VRFs, no MPLS label distribution
- Lack of scalability. VRFs on separate devices must be connected with separate circuits.
- EIGRP IPv6 VRF-Lite feature is available only in EIGRP named configurations
- ***(EIGRP) address-family ipv6 vrf <name> autonomous-system <as>***
- VRF itself does not require RD/TR to provide local routing table separation

Verify

- ***show ip vrf [id]***
- ***show ip route vrf <name> <prefix>***
- ***show ip route vrf ****
- ***{traceroute | ping} vrf ...***

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 global 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

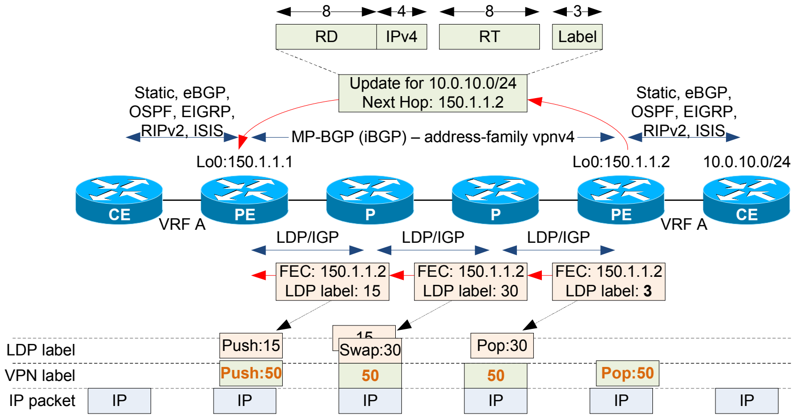
Route Target

- 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

L3 VPN

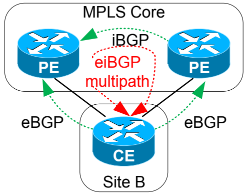
Concept

- 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
- VPN labels are exchanged between edge LSRs. They describe to which VRF packet will be sent when it reaches 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. They only need to know how to reach BGP next-hop (using IGP – usually OSPF, ISIS)
- PE routers exchange routing and label information using BGP (scalable and multi-protocol capability).

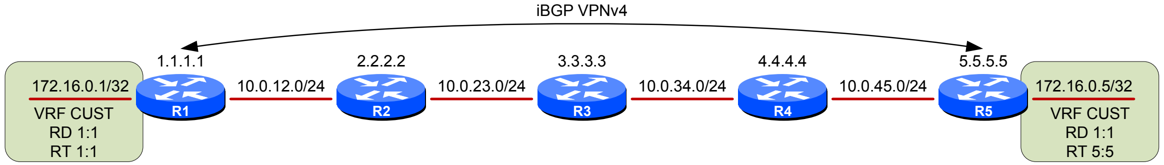


MP-BGP

- 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: 1-IPv4, 2-IPv6. SAFI: 1-ucast, 2-mcast, 4-IPv4 label forwarding, 128-labeled VPN forwarding
  - Exchanges VPNv4 MPLS VPN label (transport label)
- Address Families (same for IPv6)
  - ***(BGP) address-family vpnv4***
    iBGP prefix and label exchange between PE LSRs
  - ***(BGP) address-family ipv4 vrf <name>***
    eBGP prefix exchange between PE and CE within a VRF
  - ***(BGP) 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)
- ***(BGP) neighbor <ip> activate***
  Neighbors configured in global instance, but activated in specific family
- ***(BGP) neighbor <ip> send-community {standard | extended | both}***
  Extended communities are automaticaly exchanged if peer is activated. Use ***both*** to also send standard communities
- ***(BGP) 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***
- ***show ip bgp vpnv4 {all | rd <rd> | vrf <vrf>} ...***
- Multipath
  - Supported only by basic MPLS L3 VPNs (Inter-AS and CSC are not supported). 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



- Route Reflector
  - RR for MPLS L3 VPNs should be different than for global BGP, so potential issues can be separated
  - ***(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
  - RR is not is the data path (RR does not modify the next-hop, for which labels are exchanged and LSP is established), it only manages the control plane



RIB

```
R1#show ip route vrf CUST 172.16.0.5
Routing Table: CUST
Routing entry for 172.16.0.5/32
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 5.5.5.5 00:06:35 ago
  Routing Descriptor Blocks:
  * 5.5.5.5 (default), from 5.5.5.5, 00:06:35 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: 504
      MPLS Flags: MPLS Required
```

VRF

VPN label assigned by remote PE (VRF identifier)

Ctrl

```
R1#show bgp vpnv4 unicast all summary
[...]
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
5.5.5.5         4   100      60      59        4    0    0 00:51:56        1

R1#show bgp vpnv4 unicast all 172.16.0.5
BGP routing table entry for 1:1:172.16.0.5/32, version 4
Paths: (1 available, best #1, table CUST)
  Not advertised to any peer
  Refre
  Local                          5:5:172.16.0.5/32 (global)
    5.5.5.5 (metric 5) from 5.5.5.5 (5.5.5.5)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:5:5
      mpls labels in/out nolabel/504
      rx pathid: 0, tx pathid: 0x0
```

VPNv4 prefix received from the other PE

How to reach remote prefix (transit path to remote PE)

RT 5:5 associated with VPN label 504, assigned by the peer. RT – to which VRF import the prefix. Label – how to identify packets sent to that prefix

FIB

```
R1#show ip cef vrf CUST 172.16.0.5
172.16.0.5/32
  nexthop 10.0.12.2 GigabitEthernet0/0 label 200 504
```

200 - LDP label (transit)

504 - VPN label (bottom of the stack)

LRIB

```
R1#show mpls ldp bindings
  lib entry: 1.1.1.1/32, rev 4
        local binding:  label: imp-null
        remote binding: lsr: 2.2.2.2:0, label: 203
  lib entry: 2.2.2.2/32, rev 12
        local binding:  label: 103
        remote binding: lsr: 2.2.2.2:0, label: imp-null
  lib entry: 3.3.3.3/32, rev 10
        local binding:  label: 102
        remote binding: lsr: 2.2.2.2:0, label: 202
  lib entry: 4.4.4.4/32, rev 8
        local binding:  label: 101
        remote binding: lsr: 2.2.2.2:0, label: 201
  lib entry: 5.5.5.5/32, rev 6
        local binding:  label: 100
        remote binding: lsr: 2.2.2.2:0, label: 200
  lib entry: 10.0.12.0/24, rev 2
        local binding:  label: imp-null
        remote binding: lsr: 2.2.2.2:0, label: imp-null
  lib entry: 10.0.23.0/24, rev 13
        remote binding: lsr: 2.2.2.2:0, label: imp-null
```

LFIB

```
R1#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
100        200        5.5.5.5/32       0             Gi0/0      10.0.12.2
101        201        4.4.4.4/32       0             Gi0/0      10.0.12.2
102        202        3.3.3.3/32       0             Gi0/0      10.0.12.2
103        Pop Label  2.2.2.2/32       0             Gi0/0      10.0.12.2
104        Pop Label  172.16.0.1/32[V] 500             aggregate/CUST
```

Next hop advertised an implicit NULL

Prefix is in local L3VPN/VRF

CEF needs to do further recursion to find the L2 address. Means destination is locally connected

Vrfy

```
R1#traceroute vrf CUST 172.16.0.5 source lo10
[...]
  1 10.0.12.2 [MPLS: Labels 200/504 Exp 0] 196 msec 168 msec 184 msec
  2 10.0.23.3 [MPLS: Labels 300/504 Exp 0] 152 msec 196 msec 196 msec
  3 10.0.34.4 [MPLS: Labels 400/504 Exp 0] 216 msec 156 msec 184 msec
  4 172.16.0.5 232 msec 216 msec 168 msec
```

Transit / VPN

```
R2#show mpls ldp bindings
  lib entry: 1.1.1.1/32, rev 14
        local binding:  label: 203
        remote binding: lsr: 1.1.1.1:0, label: imp-null
        remote binding: lsr: 3.3.3.3:0, label: 303
  lib entry: 2.2.2.2/32, rev 6
        local binding:  label: imp-null
        remote binding: lsr: 1.1.1.1:0, label: 103
        remote binding: lsr: 3.3.3.3:0, label: 302
  lib entry: 3.3.3.3/32, rev 12
        local binding:  label: 202
        remote binding: lsr: 1.1.1.1:0, label: 102
        remote binding: lsr: 3.3.3.3:0, label: imp-null
  lib entry: 4.4.4.4/32, rev 10
        local binding:  label: 201
        remote binding: lsr: 1.1.1.1:0, label: 101
        remote binding: lsr: 3.3.3.3:0, label: 301
  lib entry: 5.5.5.5/32, rev 8
        local binding:  label: 200
        remote binding: lsr: 1.1.1.
        remote binding: lsr: 3.3.3.3:0, label: 300
  lib entry: 10.0.12.0/24, rev 2
        local binding:  label: imp-null
        remote binding: lsr: 1.1.1.1:0, label: imp-null
  lib entry: 10.0.23.0/24, rev 4
        local binding:  label: imp-null
        remote binding: lsr: 3.3.3.3:0, label: imp-null
  lib entry: 10.0.34.0/24, rev 15
        remote binding: lsr: 3.3.3.3:0, label: imp-null
```

Swap label 200 with 300 when sending downward to remote PE

via R3 and R4

Labels assigned by local LSR, when propagating to other LSRs

```
R2#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
200        300        5.5.5.5/32       6760          Gi1/0      10.0.23.3
201        301        4.4.4.4/32       0             Gi1/0      10.0.23.3
202        Pop Label  3.3.3.3/32       0             Gi1/0      10.0.23.3
203        Pop Label  1.1.1.1/32       6302          Gi0/0      10.0.12.1
```

Labels assigned by peers, used by local LSR

Traffic on that LSP

RIB

```
R5#show bgp vpnv4 unicast all 172.16.0.5
BGP routing table entry for 5:5:172.16.0.5/32, version 2
Paths: (1 available, best #1, table CUST)
[...]
    0.0.0.0 from 0.0.0.0 (5.5.5.5)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local, best
      Extended Community: RT:5:5
      mpls labels in/out 504/nola
      rx pathid: 0, tx pathid: 0x
```

RD – local meaning only

Local VRF name

RT and local VPN label assigned by R5, remote PE uses that label to mark the packet on the transit

```
R5#show ip route vrf CUST 172.16.0.1
Routing Table: CUST
Routing entry for 172.16.0.1/32
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 1.1.1.1 00:29:35 ago
  Routing Descriptor Blocks:
  * 1.1.1.1 (default), from 1.1.1.1, 00:29:35 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
      MPLS label: 104
      MPLS Flags: MPLS Required
```

```
R5#show bgp vpnv4 unicast all summary
[...]
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4   100      60      61        4    0    0 00:52:48        1

R5#show bgp vpnv4 unicast all 172.16.0.1
BGP routing table entry for 5:5:172.16.0.1/32, version 4
Paths: (1 available, best #1, table CUST)
  Not advertised to any peer
  Refresh Epoch 1
  Local, imported path from 1:1:172.16.0.1/32 (global)
    1.1.1.1 (metric 5) from 1.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      mpls labels in/out nolabel/104
      rx pathid: 0, tx pathid: 0x0
```

```
R5#show ip cef vrf CUST 172.16.0.1
172.16.0.1/32
  nexthop 10.0.45.4 GigabitEthernet1/0 label 403 104
```

Different VPN label as on the other side – LSP is unidirectional

```
R5#show mpls ldp bindings
  lib entry: 1.1.1.1/32, rev 13
        local binding:  label: 503
        remote binding: lsr: 4.4.4.4:0, label: 403
  lib entry: 2.2.2.2/32, rev 11
        local binding:  label: 502
        remote binding: lsr: 4.4.4.4:0, label: 402
  lib entry: 3.3.3.3/32, rev 9
        local binding:  label: 501
        remote binding: lsr: 4.4.4.4:0, label: 401
  lib entry: 4.4.4.4/32, rev 6
        local binding:  label: 500
        remote binding: lsr: 4.4.4.4:0, label: imp-null
  lib entry: 5.5.5.5/32, rev 4
        local binding:  label: imp-null
        remote binding: lsr: 4.4.4.4:0, label: 400
  lib entry: 10.0.34.0/24, rev 7
        remote binding: lsr: 4.4.4.4:0, label: imp-null
  lib entry: 10.0.45.0/24, rev 2
        local binding:  label: imp-null
        remote binding: lsr: 4.4.4.4:0, label: imp-null
```

Outside (transit) label removed (PHP), only VPN label is left

```
R5#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
500        Pop Label  4.4.4.4/32       0             Gi1/0      10.0.45.4
501        401        3.3.3.3/32       0             Gi1/0      10.0.45.4
502        402        2.2.2.2/32       0             Gi1/0      10.0.45.4
503        403        1.1.1.1/32       0             Gi1/0      10.0.45.4
504        Pop Label  172.16.0.5/32[V] 500           aggregate/CUST
```

PE-CE EIGRP

Features

- 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

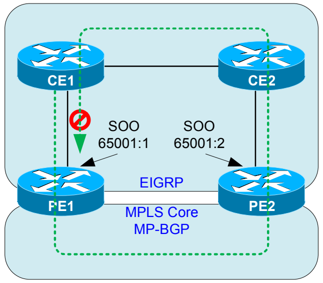
- ***router eigrp <as>***
  address-family ipv4 vrf <name>
  autonomous-system <AF AS>
  You MUST define AS for address-family even if it is the same as global AS
- ***(EIGRP) redistribute bgp <as>***
  Metric must be defined either with ***redistribite*** or with ***default-metric*** command
- ***(BGP AF) redistribute eigrp <AF AS>***
  AS must be specified even if named mode is used
- Because BGP carries vector attributes as extended communities, EIGRP can calculate feasibility conditions, so the redistributed route is seen as internal (D), not external (D EX)

Scenarios

- **1.** Sites share the same EIGRP AS – BGP carries EIGRP attributes natively. Prefixes redistributed into EIGRP seen as internal (D) with AD90 and hop count 2
- **2.** Sites share the same EIGRP AS and a backdoor link – use delay on backdor link for worse preference. SOO on a backdoor link is used as a loop prevention (only when there is high redundancy, so one site never becomes partitioned internally)
- **3.** Sites with different EIGRP ASes – BGP carries EIGRP attributes natively. Prefixes redistributed into EIGRP seen as external (D EX) with AD170 and hop count 1
- **4. Non-**EIGRP and EIRGP sites – do not use, possible loop as non-EIGRP site does not use Cost community.

SOO

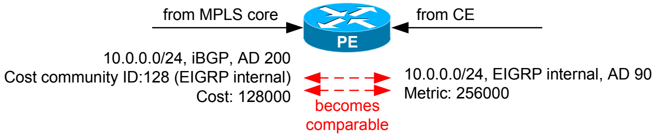
- Site of Origin – used for **loop prevention in dual-homed CE** when there is a race condition between EIGRP and BGP updates. Attached to VPNv4 route as extended community. EIGRP carries SOO as separate TLV
- SOO is added only if it is not already present. If site map matches SOO carried (in any direction) by routing update (via interface where site map is configured) the update is ignored.
- ***(IF) 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 partinoned 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 (default POI is 128). 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 (Point of Insertion) - pre-bestpath - defines when 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**
- MPLS core is transparent, does not add anything to the cost. Passed only to iBGP and confederation peers
- By default, when POI 128 is used, no BGP attributes can influence the path (even weight)
- ID is a tiebreaker when costs are the same. Lower is better. Default IDs are overwritten when redistributing into BGP, so use different ones (ex. 10) in route map. All cost communites are carried through MP-BGP. However, incomming prefix's default POI ID can be also manually overwritten via route-map on remote peer
- ***(RM) set extcommunity cost pre-bestpath 10 12345678***
  10 is less than 128, so this cost takes precedence
- ***(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
10.0.0.0/24, EIGRP internal, AD 90
Metric: 256000
becomes
comparable

```
R3#show bgp vpnv4 unicast all 192.168.0.8/32
BGP routing table entry for 100:1:192.168.0.8/32, version 13
Paths: (1 available, best #1, table CUST1)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    192.168.0.7 (metric 3) from 192.168.0.7 (192.168.0.7)
      Origin incomplete, metric 10880, localpref 1[illegible]1, best
      Extended Community: RT:100:100 Cost:pre-bestpath:128:10880
        0x8800:32768:0 0x8801:100:256 0x8802:65281:2560 0x8803:65281:1500
        0x8806:0:3232235528
      mpls labels in/out nolabel/703
      rx pathid: 0, tx pathid: 0x0
```

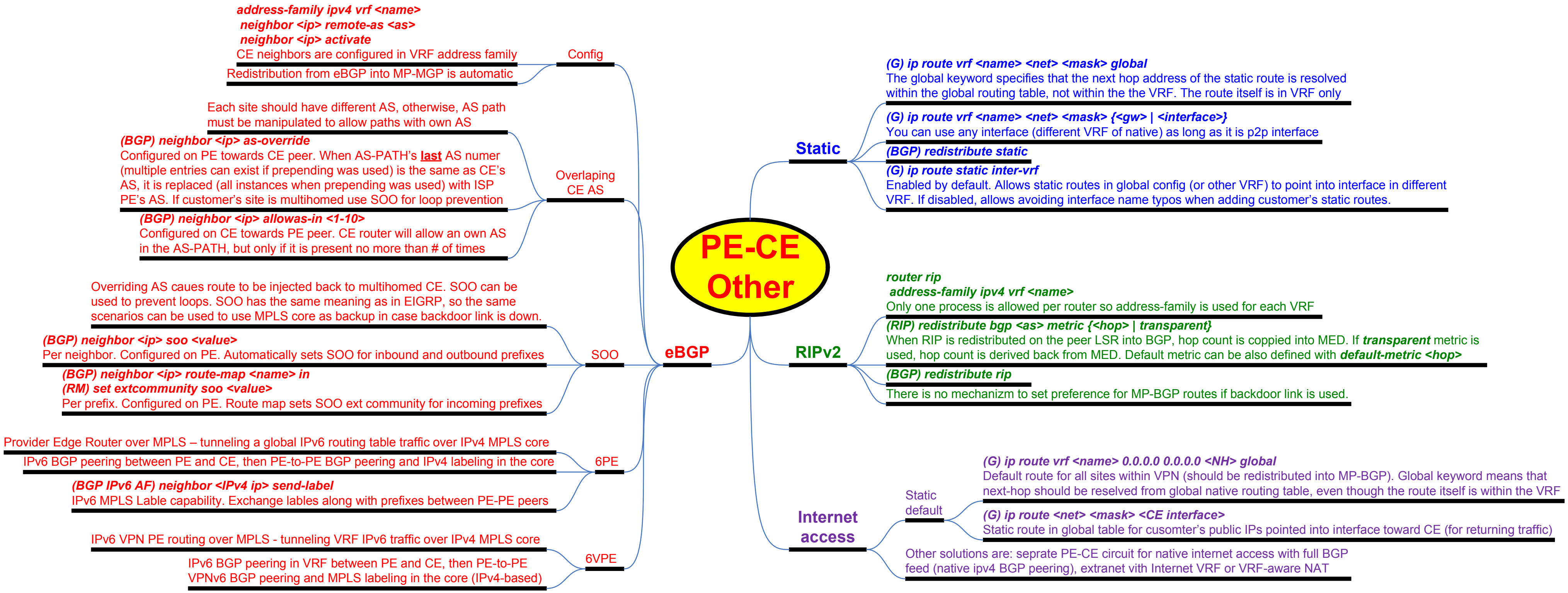
POI:Composite metric

General
0x8800 – Flags:Tag

Internal Metric if POI is 128 (absolute priority in calculations)
0x8801 – AS + Delay
0x8802 – Reliability + Hop count + BW
0x8803 – Reserved + Load + MTU

External Metric if POI is 129 (after comparing IGP cost to NH)
0x8804 – External AS + External Originator ID
0x8805 – External protocol + External Metric

Values are taken directly form the metric caluclation formula



PE-CE Other

eBGP

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

Overlaping 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 towards 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. If customer's site is multihomed use SOO for loop prevention

(BGP) neighbor <ip> allowas-in <1-10>
Configured on CE towards PE peer. CE router will allow an own AS in the AS-PATH, but only if it is present no more than # of times

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>
Per neighbor. Configured on PE. Automatically sets SOO for inbound and outbound prefixes

(BGP) neighbor <ip> route-map <name> in
(RM) set extcommunity soo <value>
Per prefix. Configured on PE. Route map sets SOO ext community for incoming prefixes

6PE

Provider Edge Router over MPLS – tunneling a global IPv6 routing table traffic over IPv4 MPLS core

IPv6 BGP peering between PE and CE, then PE-to-PE BGP peering and IPv4 labeling in the core

(BGP IPv6 AF) neighbor <IPv4 ip> send-label
IPv6 MPLS Lable capability. Exchange lables along with prefixes between PE-PE peers

6VPE

IPv6 VPN PE routing over MPLS - tunneling VRF IPv6 traffic over IPv4 MPLS core

IPv6 BGP peering in VRF between PE and CE, then PE-to-PE VPNv6 BGP peering and MPLS labeling in the core (IPv4-based)

Static

(G) ip route vrf <name> <net> <mask> global
The global keyword specifies that the next hop address of the static route is resolved within the global routing table, not within the the VRF. The route itself is in VRF only

(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 the 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

```
R3# show ip ospf 2
 Routing Process "ospf 2" with ID 10.0.13.3
[...]
 Connected to MPLS VPN Superbackbone, VRF CUST1
```

PE becomes ABR (not ASBR) – flooding boundry, even between area 0s in branches. MPLS becomes superbackbone (OSPF protocol behavior changes)

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.

Area 0 is required on PE only if there is more than one area in the same customer VRF. Non-backbone area cannot be between area 0 and superbackbone.

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)

OSPF RT:<area 4Bytes>:<route type 1Byte>:<options 1Byte>
This is NOT a Route Target, it's a Route Type, carried via MP-BGP. 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 (E2) 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

```
R3#show bgp vpnv4 unicast all 192.168.0.8
BGP routing table entry for 100:1:192.168.0.8/32, version 5
Paths: (1 available, best #1, table CUST1)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    192.168.0.7 (metric 3) from 192.168.0.7 (192.168.0.7)
      Origin incomplete, metric 2, localpref 100, valid, internal, best
      Extended Community: RT:100:100 OSPF DOMAIN ID:0x0005:0x000000020200
        OSPF RT:0.0.0.0:2:0 OSPF ROUTER ID:10.0.78.7:0
      mpls labels in/out nolabel/703
      rx pathid: 0, tx pathid: 0x0
```

Route Type

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)

SPF recalculation in one branch causes recalculations in the other area, being part of the other end of sham link

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

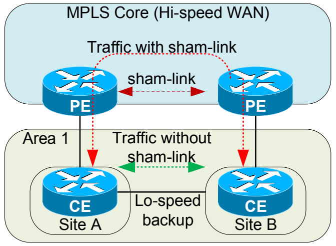
OSPF adjacency is established. LSAs are exchanged, but they are used only for path caluclations. Forwarding is still done using MP-BGP

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

(BGP) network </32 loopback> mask 255.255.255.255
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.

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

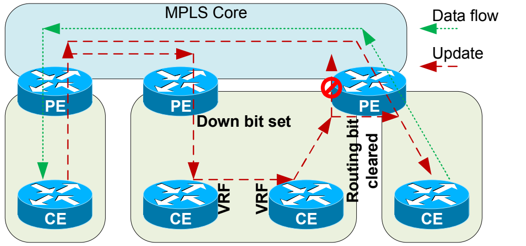
(BGP) redistribute ospf <id> match {internal | external 1 | external 2}
If match is not defined only internal routes are redistributed.

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 (downward)



Dual-homed area loop prevention

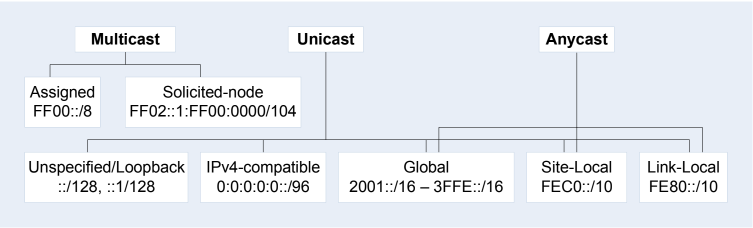
Automaticaly set in LSA 3 and 5 (only) header options field when routes are redistributed from MP-BGP into OSPF (PE to CE, but not the other way). When down bit is set for prefix received on interface which is configured with VRF, the OSPF will never use this LSA for SPF calculations. PE will not redistribute such routes back to MP-BGP

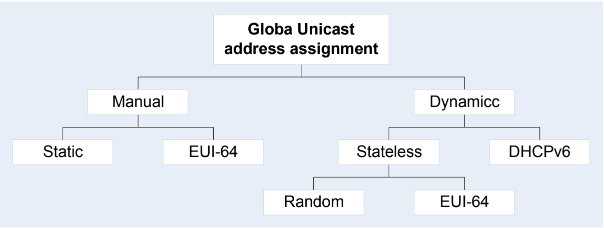
When down bit is set, routing bit gets cleared on PE. Route will not be placed into routing table even if it is the database and is the best path. Otherwise sub-optimal routing would take place (through transiting area, not mpls superbackbone)

(OSPF) capability vrf-lite
Required on CEs if VRF Lite is used (Down Bit is still set but ignored). If route is inside VRF, it will not be installed in routing table. If there is no loop danger, you can allow this route.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, and if backup link exists use tags.

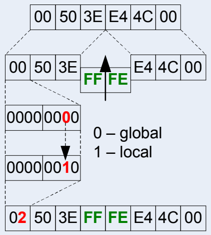
```
R3#show ip ospf database summary 192.168.0.8
[...]
  LS age: 22
  Options: (No TOS-capability, DC, Downward)
  LS Type: Summary Links(Network)
  Link State ID: 192.168.0.8 (summary Network Number)
  Advertising Router: 10.0.13.3
  LS Seq Number: 80000001
  Checksum: 0x4CE2
  Length: 28
  Network Mask: /32
        MTID: 0         Metric: 2
```

Down bit set





EUI-64 48bit MAC => 64bit EUI conversion

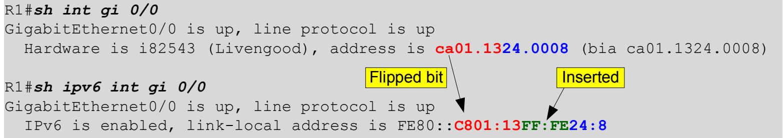


Step 1
Insert FFFE in the middle

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

```
gggg:gggg:gggg:ssss:hhhh:hhhh:hhhh:hhhh
◄──Global /48──►◄Subnet►◄────Host /64────►
2001:0000:0000:00C5:0000:0000:0000:A1B2
2001:   0:   0:  C5:    :    :    :A1B2
          2001:0:0:C5::A1B2
```

Only one leading zeros can be ommited in abbreviating IPv6 address: 2002::0:0:1, not 2002::0::1

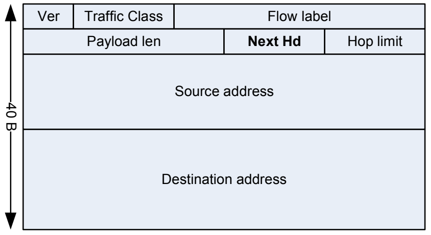

```
R1#sh int gi 0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is i82543 (Livengood), address is ca01.1324.0008 (bia ca01.1324.0008)

R1#sh ipv6 int gi 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C801:13FF:FE24:8
```

IPv6

Header

- Flow labet – identify flow to one or more end devices, still experimental
- Payload lengtd – extension headers are part of the payload, so they are counted here
- Next header – like protocol number in IPv4 (the same values). There can be 0 or more headers. Each header points to another header
  - 0: hop-by-hop options. Each router must examine this header
  - 44: feagmentation. Identification, offset, etc. Only source can fragment packets. Rouers discard IPv6 fragments
  - 60: destination options. End host must examine this header
- Hop limit – more intuitive name for TTL

IPv6 Header



Manual config

- ***(G) ipv6 unicast-routing***
- Address assignment
  - EUI-64
    - ***(IF) ipv6 address 2001:0410:0:1::/64 eui-64***
      Auto-configured from a 64-bit EUI-64 host ID (usually MAC)
    - Based on MAC has low security, as you can guess which host uses an address
    - If used on logical interface, MAC of the numerically lowest Eth is used, or the tunnel source interface's address (address will change if tunnel source changes)
  - ***(IF) ipv6 enable***
    Link-Local (only) will be configured automaticaly (host = EUI64)
  - ***(IF) ipv6 address fe80::1 link-local***
    Manualy assigned link-local address. Mask is not required, /10 is default for link-local
  - ***(IF) ipv6 address 3001:fffe::104/64 anycast***
    Anycast address
  - ***(IF) ipv6 address 2001:0410:0:1::100/64***
    Manually configured complete IPv6 address. RFC says, hosts should have /64 mask
- Link-Local addresses can overlap on interfaces of the router, they have local meaning. To ping local address use ***ping <ipv6 link-local address>%<full interface name>***
- 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
- Local host routes (L) are installed for each interface. They are seen as connected (AD 0), but they are not redistributed (***redistribute connected***). Only whole interface subnet is redistributed. Host route is only for local router – traffic to that address is processed
- General Prefix
  - Useful when using temporary addresses which will be changed in the future (change only prefix)
  - ***(G) ipv6 genral-prefix <name> <prefix>***
    ipv6 genral-prefix MY-GLOBAL 2001:A:B::/48
  - ***(IF) ipv6 address <prefix name> <host address>***
    ipv6 address MY-GLOBAL ::1/64 => 2001:A:B::1/64

Aggregatable-Global

2000::/3 – 3FFF:FFFF...FFFF

/48 provider + /16 site (subnet) + EUI-64 (intf)
3 hextets + 1 hextet + 4 hextets = 3.14 (PI) :-)

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

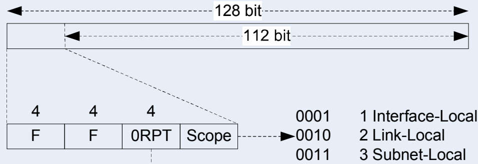
Unique Local (ULA)

Replaces Site-Local (private addresses)
FC00::/7 + EUI-64

Embeded IPv4

::/80

Multicast FF00::/8
No TTL. Scoping in address. Src address can never be Mcast.



R=1 – Embeder RP
P=1 – Based on unicast
T=1 – Temporary address
T=0 – IANA Assigned

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::D	All PIM Routers

::/128	Unspecified
::1/128	Loopback
::/0	Default

Multicast => MAC
33:33 + low-order 32 bit
FF02::1 => 33:33:00:00:00:01 MAC
Solicited node Mcast (added to each interface)
FF02::1:FFxx:xxxx/104 + LO 24bit uncst
Automaticaly created for each unicast or anycast. „ARP", DAD.

IPv6

ICMPv6

- Next-header ID: 58
- (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)

Neighbor discovery (like IPv4 ARP)

- There is no broadcast in IPv6, so no classical ARP communication
- NS for a link local address is sent to the Solicited Node Multicast FF02::1:FF00:0/104 with 24 bits set from last 24 bits of host's MAC. Host checks if it's address is unique on the segment. If so, it sends ND to FF02::1 to present itself (GARP)
- Neighbor solicitation (NS) – ICMP Code 135 – from node to node
- Neighbor advertisement (NA) – ICMP Code 136 – from nodes to a NS sender
- To get a stateless prefix or a default route host send RS to FF02::2 (all routers)
- Router solicitation (RS) – ICMP Code 133 – from nodes to all routers
- Router advertisement (RA) – ICMP Code 134 – from routers to all nodes
- ***(G) 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

```
R1#show ipv6 neighbors
IPv6 Address                    Age Link-layer Addr  State  Interface
FE80::C802:1DFF:FE91:8            0 ca02.1d91.0008   REACH  Gi0/0.123
```

Duplicate address detection (DAD)

- Host sends DAD after it is automatically assigned a global IPv6 address
- Duplicate address detection must never be performed on an anycast address
- SRC is :: (unspecified); DST is Solicited-Node for checked address
- ***(IF) ipv6 nd dad attempts <nr>***
  Default is 1. Disable - 0

Path MTU discovery

- Fragment header: 44
- Intermediate devices do NOT perform fragmentation, only end devices
- Minimum supported MTU 1280

Cache enrty states

- INCOMPLETE – the MAC address of the neighbour has not yet been determined
- REACHABLE – the neighbour is known, and reachable (recently)
- STALE – the neighbour is not known to be reachable (no recent communication)
- DELAY – delay sending probes to give other protocols a chance to provide data
- PROBE – the neighbour is no longer reachable, and unicast NS probes are sent

Routing features

- **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 (usually with keyword ***include-connected***)
- 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.
- Static to link-local address requires specyfying an interface, as the link-local address can be the same on each interface
- VRF
  - ***(G) vrf definition <name>***
    This mode is required for IPv6
  - ***(G) vrf upgrade-cli multi-af-mode ...***
  - ***(VRF) address-family ipv6***
    Must be defined for IPv6 addresses to be inside an interface VRF
  - ***(G) ipv6 route vrf <name> ...***

Stateless Address Autoconfig (SLAAC)

- Works only if router advertises /64 subnet
- NS is sent to FF02::2 by hosts just booting up. Max 3 requests to avoid flooding. RA is sent to FF02::1
- ***(IF) ipv6 nd ra suppress [all]***
  Stop sending RA (or all advertisements). RA is automaticaly enabled when global address is configured on the 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 ra suppress [all]*** – stop sending RA (or all advertisements)
- ***(IF) ipv6 nd ra lifetime <sec>***
  How long hosts should use the router as a default gateway. If set to 0, router will not advertise itself as default candidate (default 1800 sec)
- ***(IF) ipv6 nd ra interval <sec>*** - how often RA is sent (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; ***no-autoconfig*** – (A-bit) tell hosts not to use prefix for autoconfig; ***no-advertise*** – no prefix advertisement; ***at <date>*** - no adverisement after date
- ***(IF) ipv6 address autoconfig [default]***
  Configured on a client. Autoconfigures IPv6 address. Can also set a default route towards the advertising router
- ***(IF) ipv6 nd router-preference {high | medium | low}***
  Configure DRP extension to RAs in order to signal the preference value of a default router
- ***show ipv6 interface <if> prefix***
- ***show ipv6 routers*** - neighbors

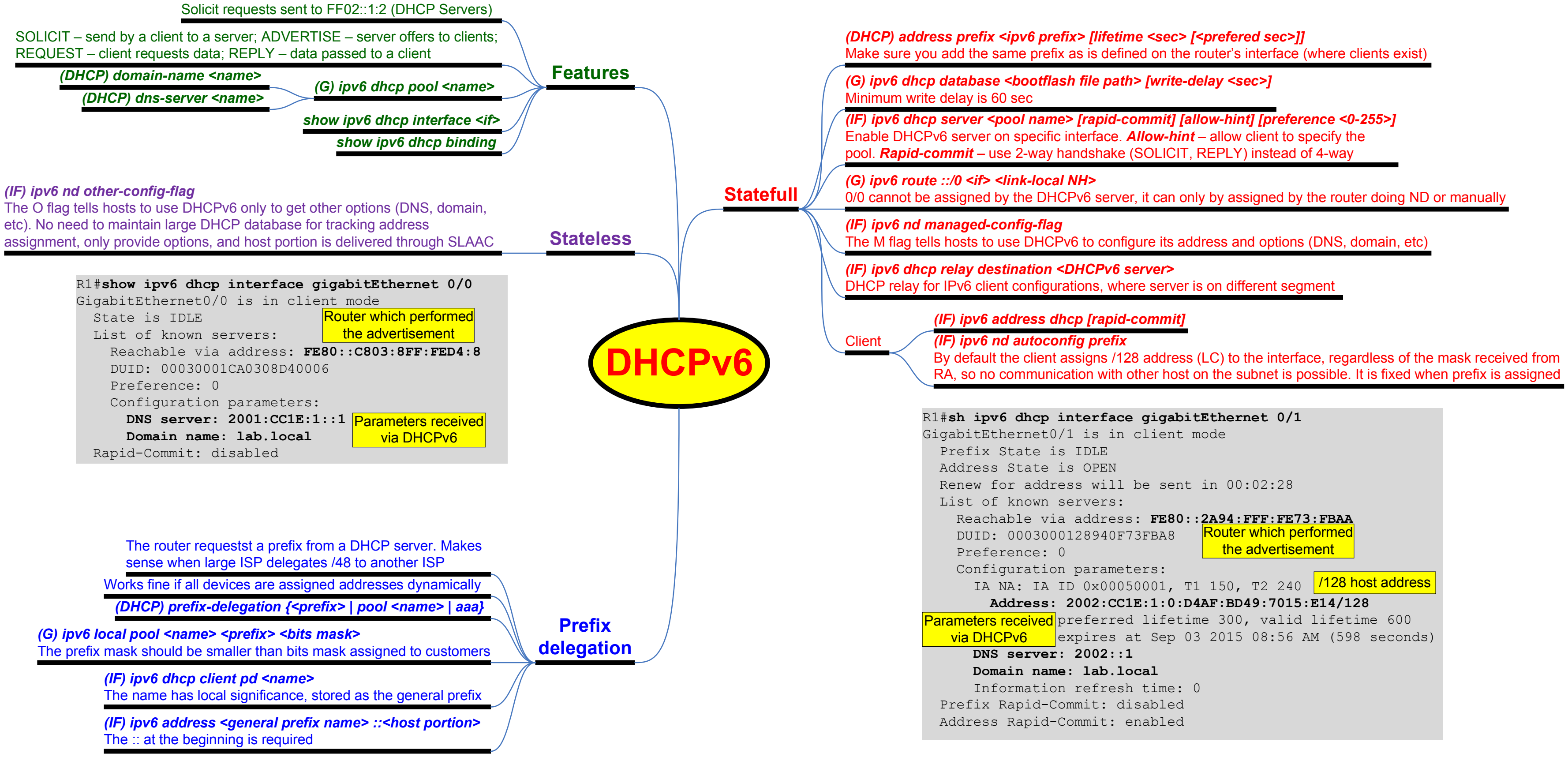
```
R2#show ipv6 interface gigabitEthernet 0/0.123
GigabitEthernet0/0.123 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C802:1DFF:FE91:8
  No Virtual link-local address(es):
  Stateless address autoconfig enabled
  Global unicast address(es):
    2001:CC1E::C802:1DFF:FE91:8, subnet is 2001:CC1E::/64 [EUI/CAL/PRE]
      valid lifetime 2591981 preferred lifetime 604781
```

Prefix received from RA

```
R2#show ipv6 route
[...]
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE - Destination
[...]
ND  ::/0 [2/0]
     via FE80::C801:1DFF:FE91:8, GigabitEthernet0/0.123
NDp 2001:CC1E::/64 [2/0]
     via GigabitEthernet0/0.123, directly connected
L   2001:CC1E::C802:1DFF:FE91:8/128 [0/0]
     via GigabitEthernet0/0.123, receive
```

Default route

Advertising router



DHCPv6

Features

Solicit requests sent to FF02::1:2 (DHCP Servers)

SOLICIT – send by a client to a server; ADVERTISE – server offers to clients; REQUEST – client requests data; REPLY – data passed to a client

(G) ipv6 dhcp pool <name>

- ***(DHCP) domain-name <name>***
- ***(DHCP) dns-server <name>***

show ipv6 dhcp interface <if>

show ipv6 dhcp binding

Stateless

(IF) ipv6 nd other-config-flag
The O flag tells hosts to use DHCPv6 only to get other options (DNS, domain, etc). No need to maintain large DHCP database for tracking address assignment, only provide options, and host portion is delivered through SLAAC

```
R1#show ipv6 dhcp interface gigabitEthernet 0/0
GigabitEthernet0/0 is in client mode
  State is IDLE
  List of known servers:
    Reachable via address: FE80::C803:8FF:FED4:8
    DUID: 00030001CA0308D40006
    Preference: 0
    Configuration parameters:
      DNS server: 2001:CC1E:1::1
      Domain name: lab.local
  Rapid-Commit: disabled
```

Router which performed the advertisement

Parameters received via DHCPv6

Statefull

(DHCP) address prefix <ipv6 prefix> [lifetime <sec> [<prefered sec>]]
Make sure you add the same prefix as is defined on the router's interface (where clients exist)

(G) ipv6 dhcp database <bootflash file path> [write-delay <sec>]
Minimum write delay is 60 sec

(IF) ipv6 dhcp server <pool name> [rapid-commit] [allow-hint] [preference <0-255>]
Enable DHCPv6 server on specific interface. ***Allow-hint*** – allow client to specify the pool. ***Rapid-commit*** – use 2-way handshake (SOLICIT, REPLY) instead of 4-way

(G) ipv6 route ::/0 <if> <link-local NH>
0/0 cannot be assigned by the DHCPv6 server, it can only by assigned by the router doing ND or manually

(IF) ipv6 nd managed-config-flag
The M flag tells hosts to use DHCPv6 to configure its address and options (DNS, domain, etc)

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

Client

- ***(IF) ipv6 address dhcp [rapid-commit]***
- ***(IF) ipv6 nd autoconfig prefix***
  By default the client assigns /128 address (LC) to the interface, regardless of the mask received from RA, so no communication with other host on the subnet is possible. It is fixed when prefix is assigned

```
R1#sh ipv6 dhcp interface gigabitEthernet 0/1
GigabitEthernet0/1 is in client mode
  Prefix State is IDLE
  Address State is OPEN
  Renew for address will be sent in 00:02:28
  List of known servers:
    Reachable via address: FE80::2A94:FFF:FE73:FBAA
    DUID: 0003000128940F73FBA8
    Preference: 0
    Configuration parameters:
      IA NA: IA ID 0x00050001, T1 150, T2 240
        Address: 2002:CC1E:1:0:D4AF:BD49:7015:E14/128
                preferred lifetime 300, valid lifetime 600
                expires at Sep 03 2015 08:56 AM (598 seconds)
      DNS server: 2002::1
      Domain name: lab.local
      Information refresh time: 0
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: enabled
```

Router which performed the advertisement

/128 host address

Parameters received via DHCPv6

Prefix delegation

The router requestst a prefix from a DHCP server. Makes sense when large ISP delegates /48 to another ISP

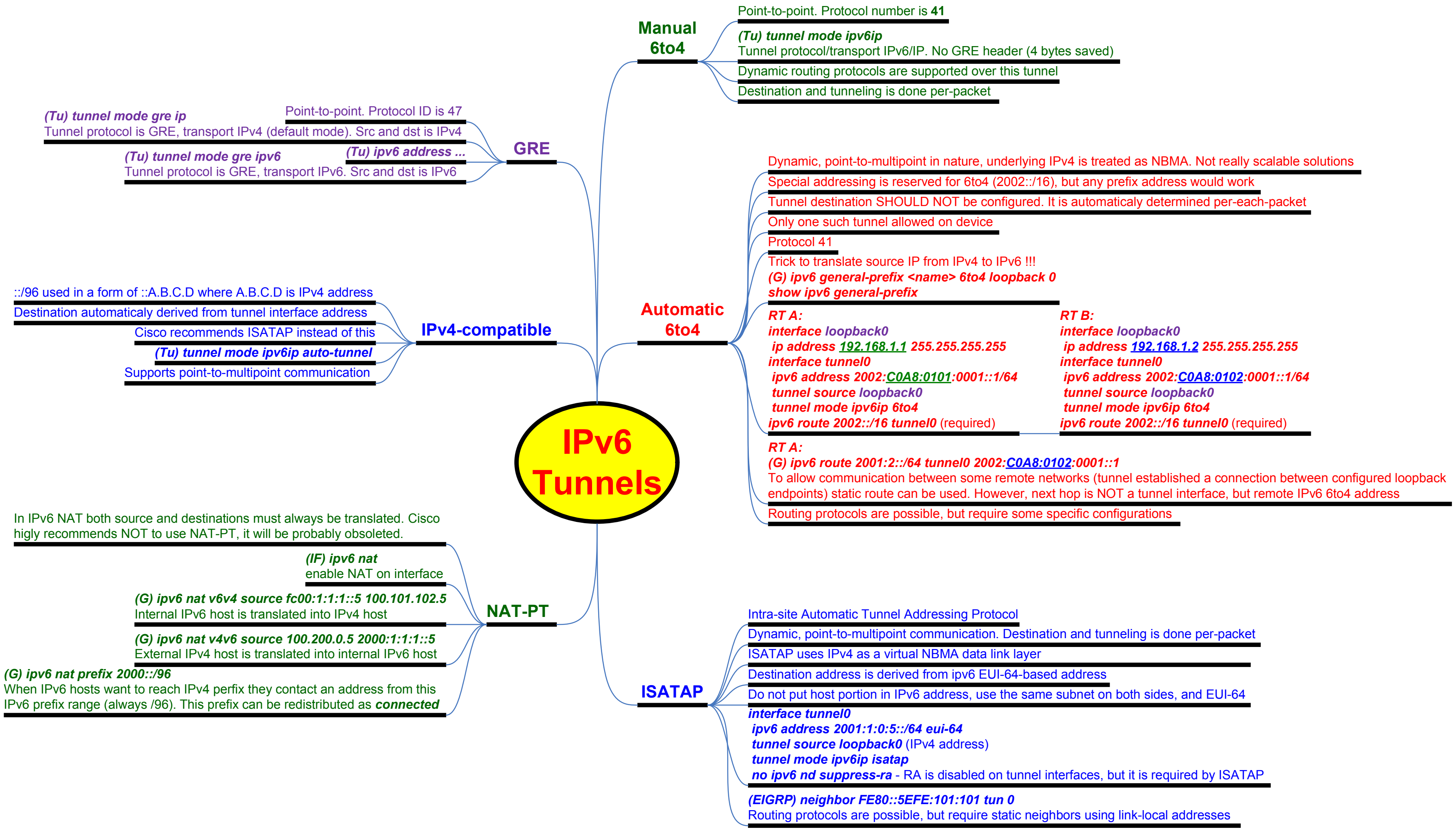
Works fine if all devices are assigned addresses dynamically

(DHCP) prefix-delegation {<prefix> | pool <name> | aaa}

(G) ipv6 local pool <name> <prefix> <bits mask>
The prefix mask should be smaller than bits mask assigned to customers

(IF) ipv6 dhcp client pd <name>
The name has local significance, stored as the general prefix

(IF) ipv6 address <general prefix name> ::<host portion>
The :: at the beginning is required



IPv6 Tunnels

Manual 6to4

- Point-to-point. Protocol number is **41**
- ***(Tu) tunnel mode ipv6ip***
  Tunnel protocol/transport IPv6/IP. No GRE header (4 bytes saved)
- Dynamic routing protocols are supported over this tunnel
- Destination and tunneling is done per-packet

GRE

- Point-to-point. Protocol ID is 47
- ***(Tu) tunnel mode gre ip***
  Tunnel protocol is GRE, transport IPv4 (default mode). Src and dst is IPv4
- ***(Tu) ipv6 address ...***
- ***(Tu) tunnel mode gre ipv6***
  Tunnel protocol is GRE, transport IPv6. Src and dst is IPv6

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
- ***(Tu) tunnel mode ipv6ip auto-tunnel***
- Supports point-to-multipoint communication

Automatic 6to4

- Dynamic, point-to-multipoint in nature, underlying IPv4 is treated as NBMA. Not really scalable solutions
- Special addressing is reserved for 6to4 (2002::/16), but any prefix address would work
- Tunnel destination SHOULD NOT be configured. It is automaticaly determined per-each-packet
- Only one such tunnel allowed on device
- Protocol 41
- 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 tunnel0 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
- Routing protocols are possible, but require some specific configurations

NAT-PT

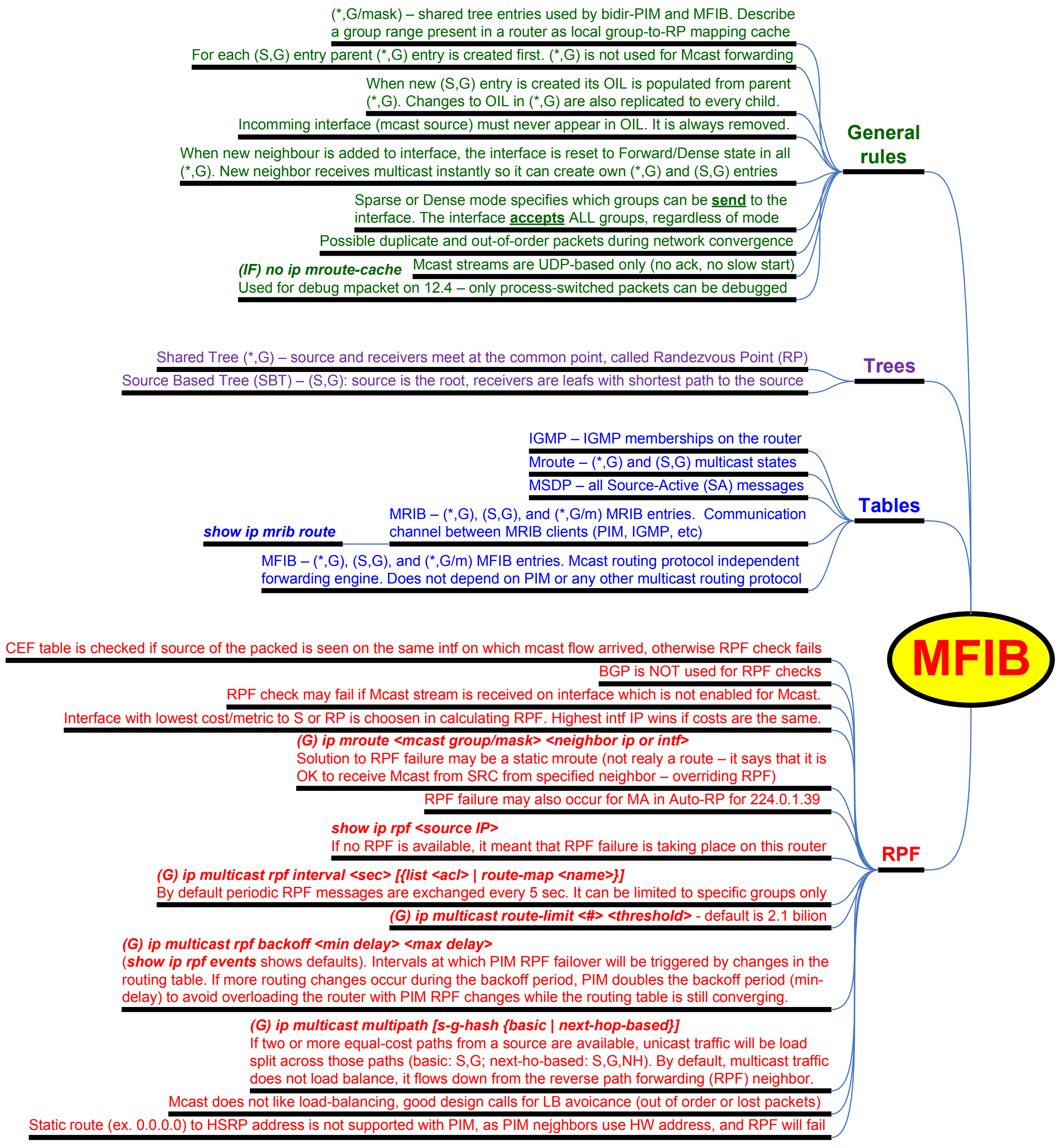
- In IPv6 NAT both source and destinations must always be translated. Cisco higly recommends NOT to use NAT-PT, it will be probably obsoleted.
- ***(IF) ipv6 nat***
  enable NAT on interface
- ***(G) ipv6 nat v6v4 source fc00:1:1:1::5 100.101.102.5***
  Internal IPv6 host is translated into IPv4 host
- ***(G) ipv6 nat v4v6 source 100.200.0.5 2000:1:1:1::5***
  External IPv4 host is translated into internal IPv6 host
- ***(G) ipv6 nat prefix 2000::/96***
  When IPv6 hosts want to reach IPv4 perfix they contact an address from this IPv6 prefix range (always /96). This prefix can be redistributed as ***connected***

ISATAP

- Intra-site Automatic Tunnel Addressing Protocol
- Dynamic, point-to-multipoint communication. Destination and tunneling is done per-packet
- ISATAP uses IPv4 as a virtual NBMA data link layer
- Destination address is derived from ipv6 EUI-64-based address
- Do not put host portion in IPv6 address, use the same subnet on both sides, and EUI-64

```
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 - RA is disabled on tunnel interfaces, but it is required by ISATAP
```

- ***(EIGRP) neighbor FE80::5EFE:101:101 tun 0***
  Routing protocols are possible, but require static neighbors using link-local addresses



MFIB

General rules

- (*,G/mask) – shared tree entries used by bidir-PIM and MFIB. Describe a group range present in a router as local group-to-RP mapping cache
- 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 (mcast source) must never appear in OIL. It is always removed.
- 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
- Possible duplicate and out-of-order packets during network convergence
- Mcast streams are UDP-based only (no ack, no slow start)
- ***(IF) no ip mroute-cache*** Used for debug mpacket on 12.4 – only process-switched packets can be debugged

Trees

- Shared Tree (*,G) – source and receivers meet at the common point, called Randezvous Point (RP)
- Source Based Tree (SBT) – (S,G): source is the root, receivers are leafs with shortest path to the source

Tables

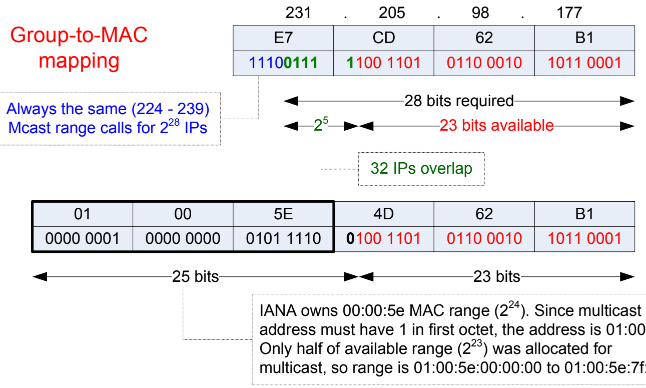
- IGMP – IGMP memberships on the router
- Mroute – (*,G) and (S,G) multicast states
- MSDP – all Source-Active (SA) messages
- MRIB – (*,G), (S,G), and (*,G/m) MRIB entries. Communication channel between MRIB clients (PIM, IGMP, etc) — ***show ip mrib route***
- MFIB – (*,G), (S,G), and (*,G/m) MFIB entries. Mcast routing protocol independent forwarding engine. Does not depend on PIM or any other multicast routing protocol

RPF

- CEF table is checked if source of the packed is seen on the same intf on which mcast flow arrived, otherwise RPF check fails
- BGP is NOT used for RPF checks
- 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.
- ***(G) ip mroute <mcast group/mask> <neighbor ip or intf>*** 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)
- 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
- ***(G) 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
- ***(G) ip multicast route-limit <#> <threshold>*** - default is 2.1 bilion
- ***(G) 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.
- ***(G) ip multicast multipath [s-g-hash {basic | next-hop-based}]*** If two or more equal-cost paths from a source are available, unicast traffic will be load split across those paths (basic: S,G; next-ho-based: S,G,NH). By default, multicast traffic does not load balance, it flows down from the reverse path forwarding (RPF) neighbor.
- Mcast does not like load-balancing, good design calls for LB avoicance (out of order or lost packets)
- Static route (ex. 0.0.0.0) to HSRP address is not supported with PIM, as PIM nejghbors use HW address, and RPF will fail

224.0.0.0 – 239.255.255.255 (1110) = 2^28
224.0.0.0/24 – Link local (TTL=1)
.1 All hosts
.2 All routers
.4 DVMRP hosts
.5 OSPF routers
.6 OSPF DR
.9 RIPv2
.10 EIGRP routers
.13 PIM routers
.12 DHCP Server/Relay Agent
.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 => A48B => 164/139
233.164.139.0/24
239.0.0.0/8 – Administrively scoped (private)

Group-to-MAC mapping



231	205	98	177
E7	CD	62	B1
11100111	1100 1101	0110 0010	1011 0001

Always the same (224 - 239) Mcast range calls for 2^{28} IPs

32 IPs overlap

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

IANA owns 00:00:5e MAC range (2^{24}). Since multicast address must have 1 in first octet, the address is 01:00:5e. Only half of available range (2^{23}) was allocated for multicast, so range is 01:00:5e:00:00:00 to 01:00:5e:7f:ff:ff

show ip mroute

D	Dense	Entry is operating in dense mode
S	Sparse	Entry is operating in sparse mode
C	Connected	Member of mcast G is directly connected
L	Local	The router is a member of a G itself
P	Pruned	Route has been pruned
R	RP-bit set	(S,G) entry has RP (usually in pruned state after STP switchover)
F	Register flag	Registered for a multicast source
T	STP-bit set	Mcast switched to STP (packets received on STP interface)
J	Joint STP	Traffic rate for STP Threshold has been reached

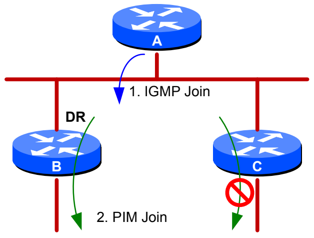
PIM

Neighbor

- Hello multicasted to 224.0.0.13 (All-PIM-Routers) as protocol 103 with TTL=1
- No sanity check. Unidirectional adjacency can be established.
- ***(IF) ip pim query-interval <sec> [msec]***
  Hello 30 sec, Hold 90 sec (3x Hello)
- PIMv2 Hello send by default, but will change to PIMv1 Query if such discovered (and back again if v1 peer disappears)
- ***(IF) ip pim passive***
  No PIM messages are sent nor accepted. IF becomes DR/DF (always). Use on LANs with single router, otherwise duplicate traffic or loop occurs (BiDir)
- ***(IF) ip pim neighbor-filter <acl>***
  Filter PIM messages received from specified peers (standard ACL)
- PIM does not announce any routes, relies on underlying IGP

Designated Router

- Elected on every shared segment
- ***(IF) ip pim dr-priority <#>***
  Higest Priority (default 1) or IP. New router with higher priority/IP preempts existing DR
- Used mainly for IGMPv1 (querier). 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.
- ***(IF) ip pim redundancy <HSRP group> dr-priority <#>***
  Bind PIM DR to active HSRP router. Priority must be larger than non-redundancy DR priority (so min. value is 2). The name is taken from ***standby <#> name***



Snooping

- Switch restricts mcast packets for each mcast group to mcast router ports that have downstream receivers joined to that group (default is flood traffic on all router ports)
- The AUTO-RP groups (224.0.1.39 and 224.0.1.40) are always flooded
- ***(G/IF) ip pim snooping***
- IGMP snooping must be also enabled
- Either RGMP or PIM snooping can be enabled in a VLAN but not both
- ***(G) no ip pim snooping dr-flood***
  Enabled by default. Use on switches that have no DRs attached

PIM DM

Rules

- Based on source tree (shortest-path tree SPT) - always
- Flood and prune algorithm. Implicit join (push)
- OIL 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

Proxy

- ***(IF) ip pim dense-mode proxy-register***
  Connect dense region to sparse region. Register-rate-limit is set to 2/sec (possibly large number of sources from dense regions)
- DR is responsible for proxy-registering

Graft

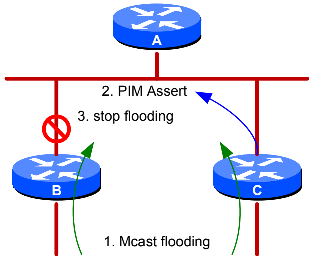
- Speeds up convergence, without waiting for periodic re-flooding (3 min Prune timer)
- Joining STP when a LAN client joins with IGMP

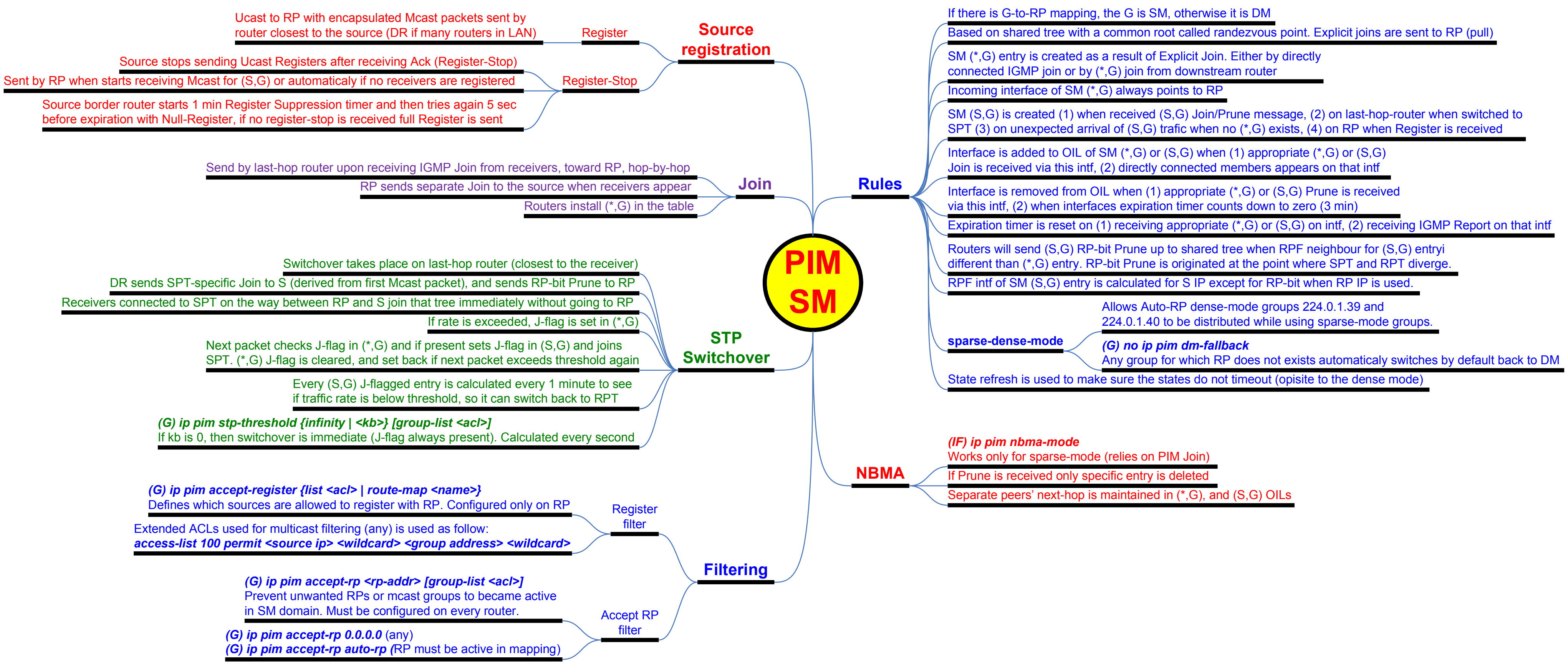
Pruning

- 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
- (S, G) entries remain in table after prunning, although traffic stops flowing on proned interfaces
- Prune-override – upstream router receiving Prune from downstream router waits 3 sec for possible Join from another router on a shared LAN. The other router hears Prune message and re-sends PIM Join as an override
- State refresh
  - Keepalive sent from the root of STP (closest to the source) to see if downstream routers still DON'T want to receive traffic
  - No need to reflood on unneded segments and wait for Prune
  - (S,G) state is still kept
  - ***(G) ip pim state-refresh disable***
    State-refresh is enabled by default
  - ***(IF) ip pim state-refresh origination-interval <sec>***
    Define origination of the PIM DM State Refresh control message (60 sec default)

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 to the source wins
  - 3. If metric is the same the highest IP is a tie-breaker
- ***show ip mroute <mcast addr>***
  Incoming interface RPF neighbor marked with *





PIM SM

Source registration

- Register
  - Ucast to RP with encapsulated Mcast packets sent by router closest to the source (DR if many routers in LAN)
- Register-Stop
  - Source stops sending Ucast Registers after receiving Ack (Register-Stop)
  - 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

Join

- Send by last-hop router upon receiving IGMP Join from receivers, toward RP, hop-by-hop
- RP sends separate Join to the source when receivers appear
- Routers install (*,G) in the table

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 checks 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 again
- 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
- ***(G) ip pim stp-threshold {infinity | <kb>} [group-list <acl>]***
  If kb is 0, then switchover is immediate (J-flag always present). Calculated every second

Filtering

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

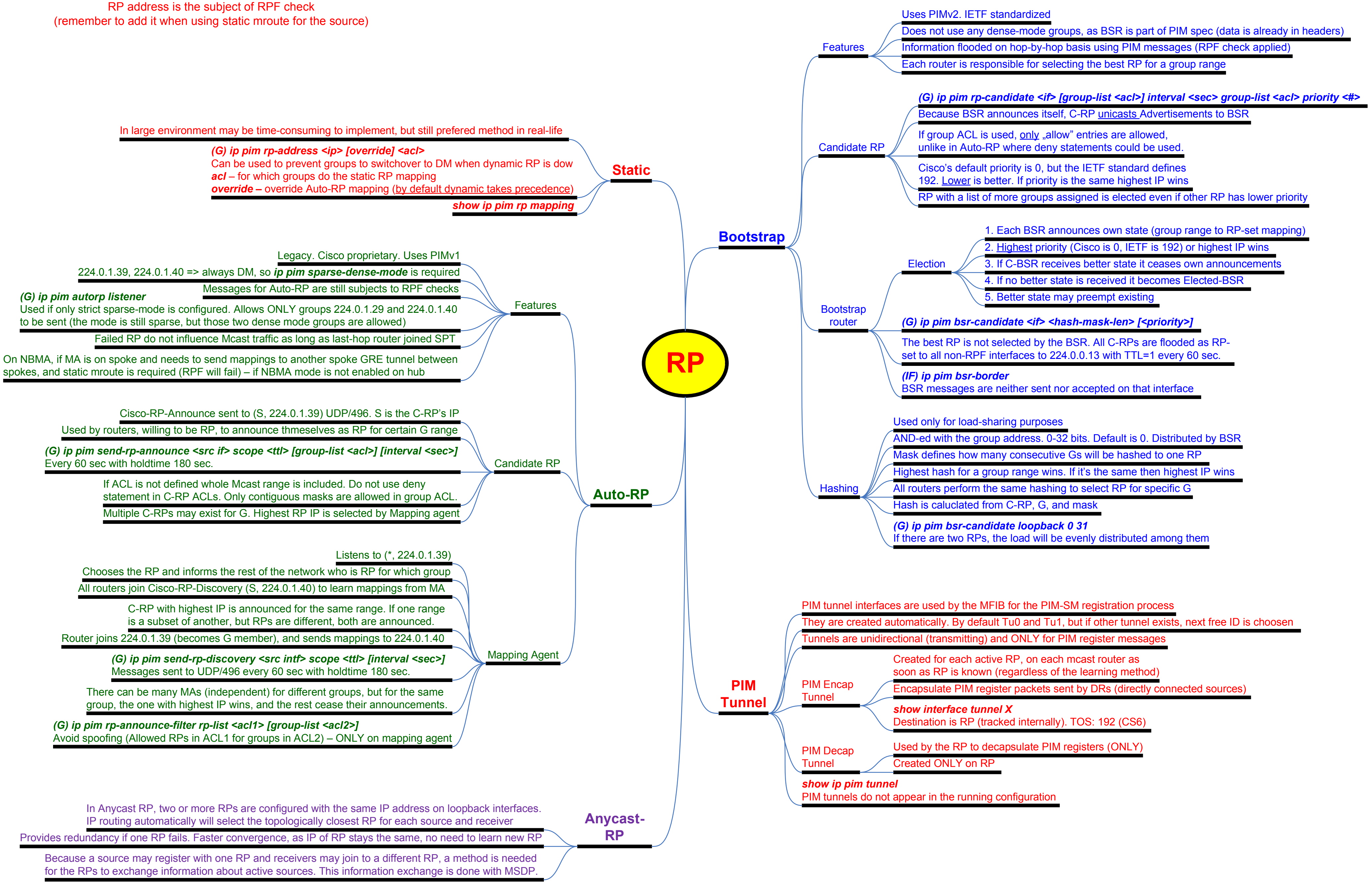
Rules

- If there is G-to-RP mapping, the G is SM, otherwise it is DM
- Based on shared tree with a common root called randezvous point. Explicit joins are sent to RP (pull)
- 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.
  - ***(G) no ip pim dm-fallback***
    Any group for which RP does not exists automaticaly switches by default back to DM
- State refresh is used to make sure the states do not timeout (opisite to the dense mode)

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

RP address is the subject of RPF check
(remember to add it when using static mroute for the source)



RP

Static

- In large environment may be time-consuming to implement, but still prefered method in real-life
- ***(G) ip pim rp-address <ip> [override] <acl>***
  Can be used to prevent groups to switchover to DM when dynamic RP is dow
  acl – for which groups do the static RP mapping
  override – override Auto-RP mapping (by default dynamic takes precedence)
- ***show ip pim rp mapping***

Auto-RP

Features

- Legacy. Cisco proprietary. Uses PIMv1
- 224.0.1.39, 224.0.1.40 => always DM, so ***ip pim sparse-dense-mode*** is required
- Messages for Auto-RP are still subjects to RPF checks
- ***(G) ip pim autorp listener***
  Used if only strict 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

Candidate RP

- Cisco-RP-Announce sent to (S, 224.0.1.39) UDP/496. S is the C-RP's IP
- Used by routers, willing to be RP, to announce thmeselves as RP for certain G range
- ***(G) ip pim send-rp-announce <src if> scope <ttl> [group-list <acl>] [interval <sec>]***
  Every 60 sec with holdtime 180 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 RP IP is selected by Mapping agent

Mapping Agent

- Listens to (*, 224.0.1.39)
- Chooses the RP and informs the rest of the network who is RP for which group
- All routers join Cisco-RP-Discovery (S, 224.0.1.40) to learn mappings from MA
- 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
- ***(G) ip pim send-rp-discovery <src intf> scope <ttl> [interval <sec>]***
  Messages sent to UDP/496 every 60 sec with holdtime 180 sec.
- There can be many MAs (independent) for different groups, but for the same group, the one with highest IP wins, and the rest cease their announcements.
- ***(G) 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

Anycast-RP

- 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
- Provides redundancy if one RP fails. Faster convergence, as IP of RP stays the same, no need to learn new RP
- 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.

Bootstrap

Features

- Uses PIMv2. IETF standardized
- Does not use any dense-mode groups, as BSR is part of PIM spec (data is already in headers)
- 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

- ***(G) 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 assigned 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
- ***(G) 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

- Used only for load-sharing purposes
- AND-ed with the group address. 0-32 bits. Default is 0. Distributed by BSR
- 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
- ***(G) ip pim bsr-candidate loopback 0 31***
  If there are two RPs, the load will be evenly distributed among them

PIM Tunnel

- PIM tunnel interfaces are used by the MFIB for the PIM-SM registration process
- They are created automatically. By default Tu0 and Tu1, but if other tunnel exists, next free ID is choosen
- Tunnels are unidirectional (transmitting) and ONLY for PIM register messages
- PIM Encap Tunnel
  - Created for each active RP, on each mcast router as soon as RP is known (regardless of the learning method)
  - Encapsulate PIM register packets sent by DRs (directly connected sources)
  - ***show interface tunnel X***
    Destination is RP (tracked internally). TOS: 192 (CS6)
- PIM Decap Tunnel
  - Used by the RP to decapsulate PIM registers (ONLY)
  - Created ONLY on RP
- ***show ip pim tunnel***
  PIM tunnels do not appear in the running configuration

PIM Other

BiDir

- Many to many, receivers are also senders. Traffic may flow up and down the tree
- Based only on shared tree (RPT). No switching to SPT
- Source sends traffic unconditionaly to RP at any time (no PIM Register process like in SM, so no PIM DRs exist)
- Designated Forwarder
  - Used on each link for loop prevention, like PIM assert (RPF check schema changes)
  - Lowest metric to RP or highest IP wins
  - Only DF can forward traffic upstream (to RP), all other devices are downstream facing
  - ***show ip pim interface df*** – winner does not have a * in the output
- No (S,G) entries, only (*,G) mroute states are active towards RP
- ***(G) ip pim bidir-enable***
  All routers must agree on BiDir or loop occurs. BiDir does not use RPF checks
- 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. PIM Join sent toward the source
- Only edge routers must support SSM, other routers only require PIM-SM
- ***(IF) ip igmp version 3***
  Requires IGMPv3 (INCLUDE/EXCLUDE messages). Hosts can decide which sources they want to join explicitly. The (*,G) joins are dropped.
- ***(G) 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

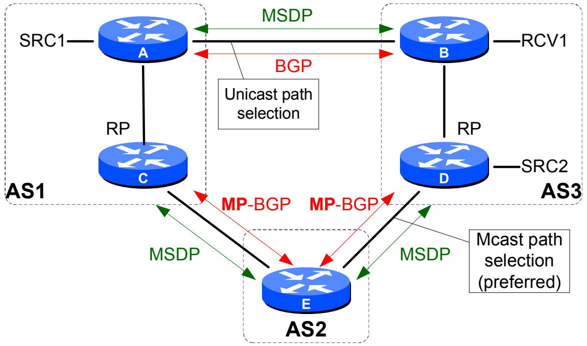
MSDP

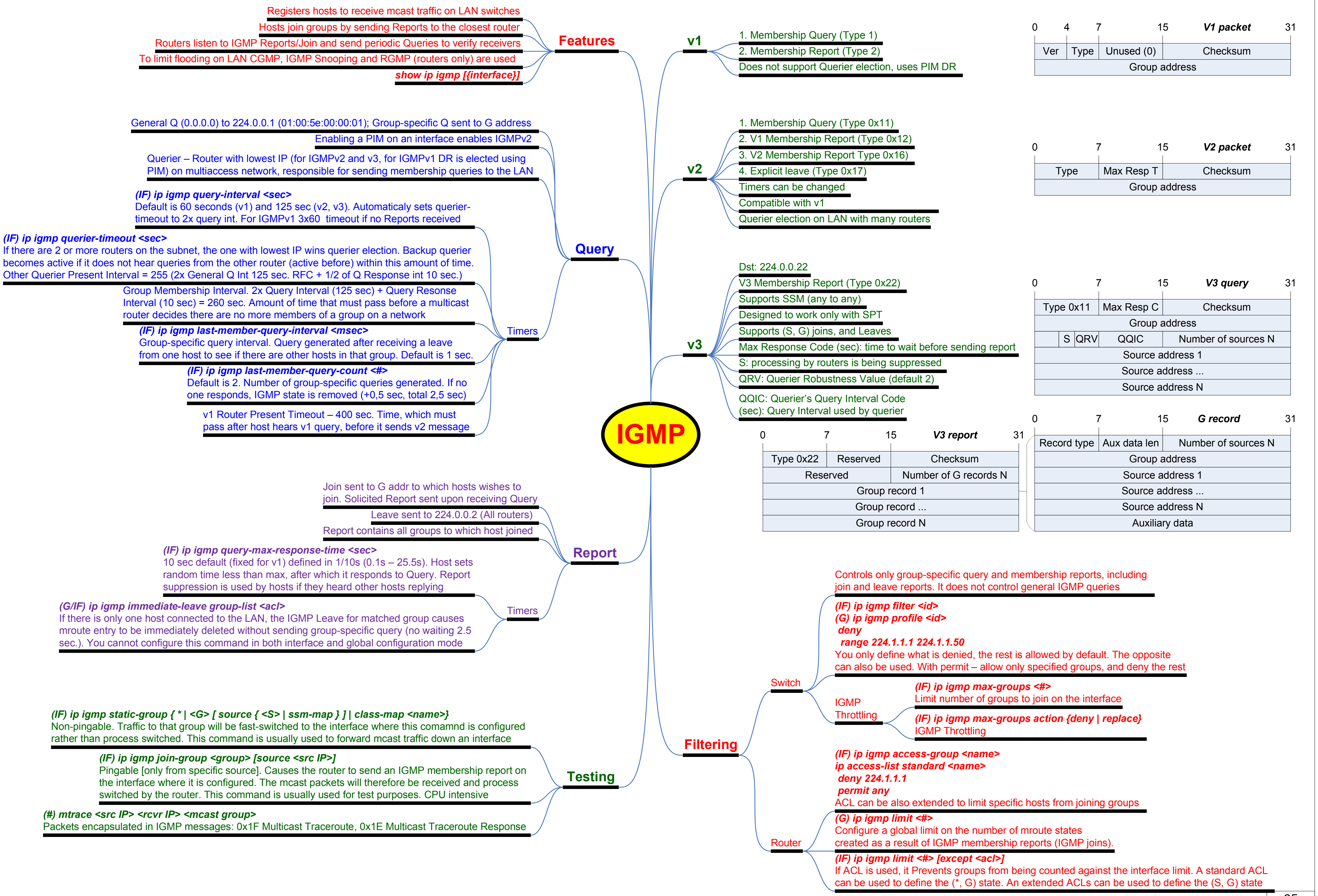
Features

- Standard-based protocol. Still requires PIM for building trees
- MSDP allows multicast sources for a group to be known to all RPs in different domains
- Does not require MP-BGP, but in real-life heavily depends on it
- RP runs MSDP over the TCP/639 to discover multicast sources in other domains
- No (S, G) states are created untill PIM Join is received (MSDP is only a control plane)
- 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)
- SA messages are forwarded only after RPF check is performed based on RP IP address
- ***(G) ip msdp originator-id <intf>***
- The MSDP device forwards the message to all MSDP peers other than the RPF peer
- ***(G) ip msdp peer <ip> connect-source <if> [remote-as <as>]***
  Configured on RP. Source must be the same as BGP source
- For Anycast-RP the MSDP peering address must be different than the Anycast RP address (TCP session must be established)
- ***(#) ip msdp sa-request <peer IP>***
  Request immediate SA data, without waiting for periodic messages
- ***(G) ip msdp mesh-group <name> <peer IP>***
  Do not send SA messages to other peers in the same group (SA messages are reduced). Peers must be connected in full mesh. All peers must be in the same group (name)
- ***show ip msdp {peer | count | sa-cache}***

MP-BGP

- Changes RPF check rules for mcast traffic (advertises networks where **sources**, not receivers reside)
- MP-BGP is preferred over unicast protocols for RPF check (like mroute, but dynamic)
- Neighbors must agree on address-family negotiated. All BGP rules apply
- ***(BGP) address-family ipv4 multicast***
- ***(AF) neighbor <ip> activate***
- ***(AF) network <net> mask <mask>***
  Advertise source networks
- ***show ip ipv4 multicast summary***





IGMP

Features

- Registers hosts to receive mcast traffic on LAN switches
- Hosts join groups by sending Reports to the closest router
- Routers listen to IGMP Reports/Join and send periodic Queries to verify receivers
- To limit flooding on LAN CGMP, IGMP Snooping and RGMP (routers only) are used
- ***show ip igmp [{interface}]***

Query

- General Q (0.0.0.0) to 224.0.0.1 (01:00:5e:00:00:01); Group-specific Q sent to G address
- Enabling a PIM on an interface enables IGMPv2
- Querier – Router with lowest IP (for IGMPv2 and v3, for IGMPv1 DR is elected using PIM) on multiaccess network, responsible for sending membership queries to the LAN
- Timers
  - ***(IF) ip igmp query-interval <sec>***
    Default is 60 seconds (v1) and 125 sec (v2, v3). Automaticaly sets querier-timeout to 2x query int. For IGMPv1 3x60 timeout if no Reports received
  - ***(IF) ip igmp querier-timeout <sec>***
    If there are 2 or more routers on the subnet, the one with lowest IP wins querier election. Backup querier becomes active if it does not hear queries from the other router (active before) within this amount of time. Other Querier Present Interval = 255 (2x General Q Int 125 sec. RFC + 1/2 of Q Response int 10 sec.)
  - Group Membership Interval. 2x Query Interval (125 sec) + Query Resonse Interval (10 sec) = 260 sec. Amount of time that must pass before a multicast router decides there are no more members of a group on a network
  - ***(IF) ip igmp last-member-query-interval <msec>***
    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. If no one responds, IGMP state is removed (+0,5 sec, total 2,5 sec)
  - v1 Router Present Timeout – 400 sec. Time, which must pass after host hears v1 query, before it sends v2 message

Report

- Join sent to G addr to which hosts wishes to join. Solicited Report sent upon receiving Query
- 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 <sec>***
    10 sec default (fixed for v1) defined in 1/10s (0.1s – 25.5s). Host sets random time less than max, after which it responds to Query. Report suppression is used by hosts if they heard other hosts replying
  - ***(G/IF) 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 without sending group-specific query (no waiting 2.5 sec.). You cannot configure this command in both interface and global configuration mode

Testing

- ***(IF) ip igmp static-group { * | <G> [source { <S> | ssm-map }] | class-map <name>}***
  Non-pingable. Traffic to that group will be fast-switched to the interface where this comamnd is configured rather than process switched. This command is usually used to forward mcast traffic down an interface
- ***(IF) ip igmp join-group <group> [source <src IP>]***
  Pingable [only from specific source]. Causes the router to send an IGMP membership report on the interface where it is configured. The mcast packets will therefore be received and process switched by the router. This command is usually used for test purposes. CPU intensive
- ***(#) mtrace <src IP> <rcvr IP> <mcast group>***
  Packets encapsulated in IGMP messages: 0x1F Multicast Traceroute, 0x1E Multicast Traceroute Response

v1

- 1. Membership Query (Type 1)
- 2. Membership Report (Type 2)
- Does not support Querier election, uses PIM DR

V1 packet

0	4	7	15	31
Ver	Type	Unused (0)	Checksum	
Group address				

v2

- 1. Membership Query (Type 0x11)
- 2. V1 Membership Report (Type 0x12)
- 3. V2 Membership Report Type 0x16)
- 4. Explicit leave (Type 0x17)
- Timers can be changed
- Compatible with v1
- Querier election on LAN with many routers

V2 packet

0	7	15	31
Type	Max Resp T	Checksum	
Group address			

v3

- Dst: 224.0.0.22
- V3 Membership Report (Type 0x22)
- Supports SSM (any to any)
- Designed to work only with SPT
- Supports (S, G) joins, and Leaves
- Max Response Code (sec): time to wait before sending report
- S: processing by routers is being suppressed
- QRV: Querier Robustness Value (default 2)
- QQIC: Querier's Query Interval Code (sec): Query Interval used by querier

V3 query

0			7	15	31
Type 0x11			Max Resp C	Checksum	
Group address					
	S	QRV	QQIC	Number of sources N	
Source address 1					
Source address ...					
Source address N					

V3 report

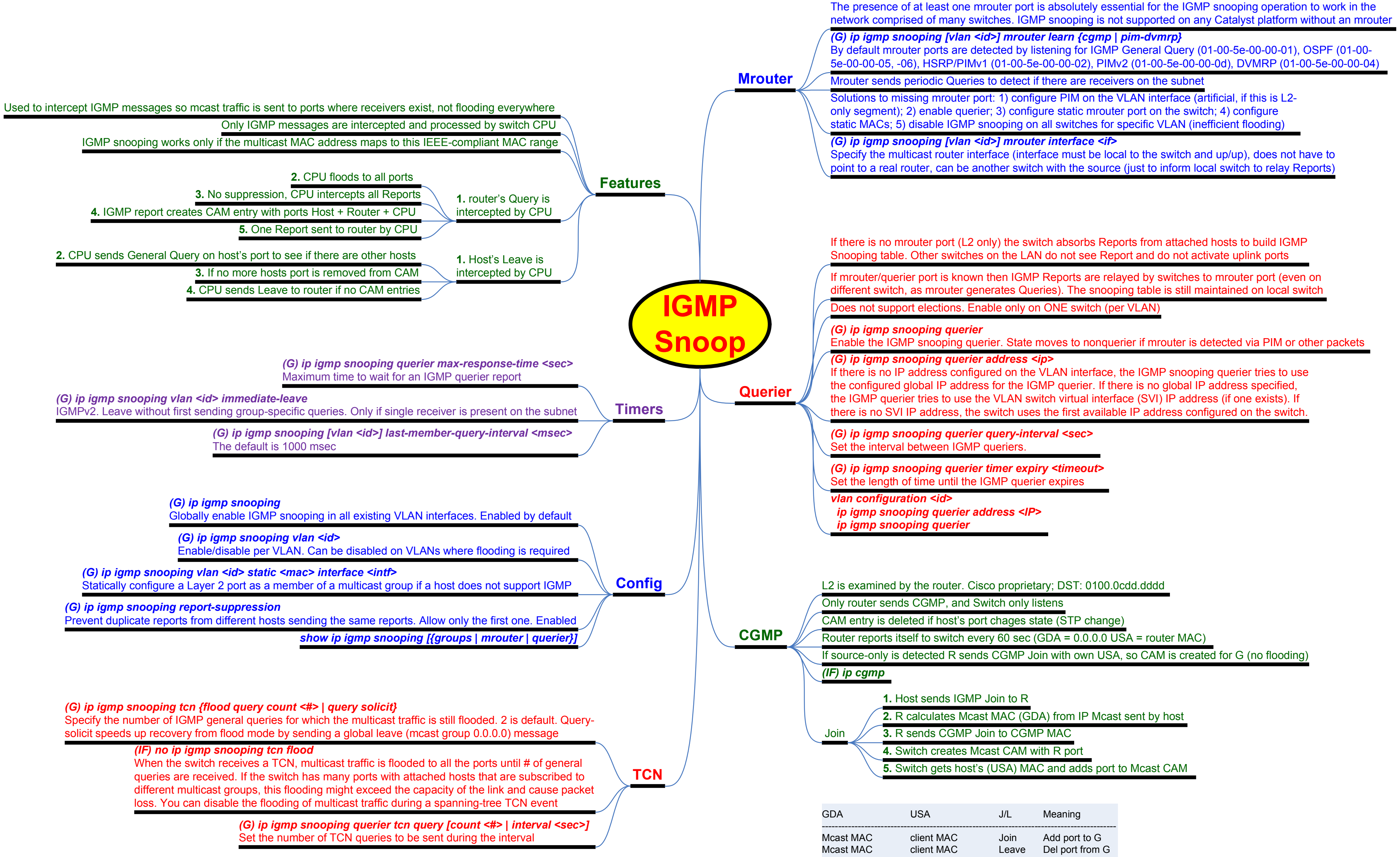
0	7	15	31
Type 0x22	Reserved	Checksum	
Reserved		Number of G records N	
Group record 1			
Group record ...			
Group record N			

G record

0	7	15	31
Record type	Aux data len	Number of sources N	
Group address			
Source address 1			
Source address ...			
Source address N			
Auxiliary data			

Filtering

- Switch
  - Controls only group-specific query and membership reports, including join and leave reports. It does not control general IGMP queries
  - ***(IF) ip igmp filter <id>***
    (G) 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. The opposite can also be used. With permit – allow only specified groups, and deny the rest
  - IGMP Throttling
    - ***(IF) ip igmp max-groups <#>***
      Limit number of groups to join on the interface
    - ***(IF) ip igmp max-groups action {deny | replace}***
      IGMP Throttling
- 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



IGMP Snoop

Features

Used to intercept IGMP messages so mcast traffic is sent to ports where receivers exist, not flooding everywhere

Only IGMP messages are intercepted and processed by switch CPU

IGMP snooping works only if the multicast MAC address maps to this IEEE-compliant MAC range

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

Timers

(G) ip igmp snooping querier max-response-time <sec>
Maximum time to wait for an IGMP querier report

(G) ip igmp snooping vlan <id> immediate-leave
IGMPv2. Leave without first sending group-specific queries. Only if single receiver is present on the subnet

(G) ip igmp snooping [vlan <id>] last-member-query-interval <msec>
The default is 1000 msec

Config

(G) ip igmp snooping
Globally enable IGMP snooping in all existing VLAN interfaces. Enabled by default

(G) ip igmp snooping vlan <id>
Enable/disable per VLAN. Can be disabled on VLANs where flooding is required

(G) ip igmp snooping vlan <id> static <mac> interface <intf>
Statically configure a Layer 2 port as a member of a multicast group if a host does not support IGMP

(G) ip igmp snooping report-suppression
Prevent duplicate reports from different hosts sending the same reports. Allow only the first one. Enabled

show ip igmp snooping [{groups | mrouter | querier}]

TCN

(G) ip igmp snooping tcn {flood query count <#> | query solicit}
Specify the number of IGMP general queries for which the multicast traffic is still flooded. 2 is default. Query-solicit speeds up recovery from flood mode by sending a global leave (mcast group 0.0.0.0) message

(IF) no ip igmp snooping tcn flood
When the switch receives a TCN, multicast traffic is flooded to all the ports until # of general queries are received. If the switch has many ports with attached hosts that are subscribed to different multicast groups, this flooding might exceed the capacity of the link and cause packet loss. You can disable the flooding of multicast traffic during a spanning-tree TCN event

(G) ip igmp snooping querier tcn query [count <#> | interval <sec>]
Set the number of TCN queries to be sent during the interval

Mrouter

The presence of at least one mrouter port is absolutely essential for the IGMP snooping operation to work in the network comprised of many switches. IGMP snooping is not supported on any Catalyst platform without an mrouter

(G) ip igmp snooping [vlan <id>] mrouter learn {cgmp | pim-dvmrp}
By default mrouter ports are detected by listening for IGMP General Query (01-00-5e-00-00-01), OSPF (01-00-5e-00-00-05, -06), HSRP/PIMv1 (01-00-5e-00-00-02), PIMv2 (01-00-5e-00-00-0d), DVMRP (01-00-5e-00-00-04)

Mrouter sends periodic Queries to detect if there are receivers on the subnet

Solutions to missing mrouter port: 1) configure PIM on the VLAN interface (artificial, if this is L2-only segment); 2) enable querier; 3) configure static mrouter port on the switch; 4) configure static MACs; 5) disable IGMP snooping on all switches for specific VLAN (inefficient flooding)

(G) ip igmp snooping [vlan <id>] mrouter interface <if>
Specify the multicast router interface (interface must be local to the switch and up/up), does not have to point to a real router, can be another switch with the source (just to inform local switch to relay Reports)

Querier

If there is no mrouter port (L2 only) the switch absorbs Reports from attached hosts to build IGMP Snooping table. Other switches on the LAN do not see Report and do not activate uplink ports

If mrouter/querier port is known then IGMP Reports are relayed by switches to mrouter port (even on different switch, as mrouter generates Queries). The snooping table is still maintained on local switch

Does not support elections. Enable only on ONE switch (per VLAN)

(G) ip igmp snooping querier
Enable the IGMP snooping querier. State moves to nonquerier if mrouter is detected via PIM or other packets

(G) ip igmp snooping querier address <ip>
If there is no IP address configured on the VLAN interface, the IGMP snooping querier tries to use the configured global IP address for the IGMP querier. If there is no global IP address specified, the IGMP querier tries to use the VLAN switch virtual interface (SVI) IP address (if one exists). If there is no SVI IP address, the switch uses the first available IP address configured on the switch.

(G) ip igmp snooping querier query-interval <sec>
Set the interval between IGMP queriers.

(G) ip igmp snooping querier timer expiry <timeout>
Set the length of time until the IGMP querier expires

vlan configuration <id>
 ip igmp snooping querier address <IP>
 ip igmp snooping querier

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)

(IF) 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

Mcast

MVR

Multicast VLAN registration intercepts IGMP Joins

Designed for applications using wide-scale deployment of multicast traffic across an Ethernet ring-based SP network

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

(G) mvr group <ip> [<count>]
Enbale MVR for a group or # of consecutive groups (max 256). Groups should not be aliasing (32:1 ratio)

show mvr

(G) mvr mode {dynamic | compatible}
Default mode is compatible, which requires static IGMP snooping entries

(G) mvr vlan <id>
Define which VLAN carries actual multicast traffic

(IF) mvr type {source | receiver}
Define source and receiver interfaces

If IGMP snooping and MVR are both enabled, MVR reacts only to join and leave messages from multicast groups configured under MVR. Join and leave messages from all other multicast groups are managed by IGMP snooping

In compatible mode, multicast data received by MVR hosts is forwarded to all MVR data ports, regardless of MVR host membership on those ports. In dynamic mode, multicast data received by MVR hosts on the switch is forwarded from only those MVR data and client ports that the MVR hosts have joined, either by IGMP reports or by MVR static configuration

(G) mvr querytime value
Define the maximum time to wait for IGMP report memberships on a receiver port before removing the port from multicast group membership. The value is in tenths of a second. The range is 1 to 100, and the default is 5 tenths or one-half second.

(IF) mvr vlan <id> group [<ip>]
Statically configure a port to receive multicast traffic sent to the multicast VLAN and the IP multicast address. A port statically configured as a member of a group remains a member of the group until statically removed. In compatible mode, this command applies to only receiver ports. In dynamic mode, it applies to receiver ports and source ports.

(IF) mvr immediate
This command applies to only receiver ports and should only be enabled on receiver ports to which a single receiver device is connected.

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

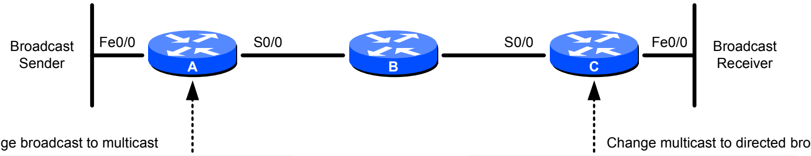
(IF) ip multicast boundry <acl> [filter-autorp]
access-list <acl> deny 224.0.1.39
access-list <acl> deny 224.0.1.40
access-list <acl> permit 224.0.0.0 15.255.255.255

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.

Multicast helper for bcast 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

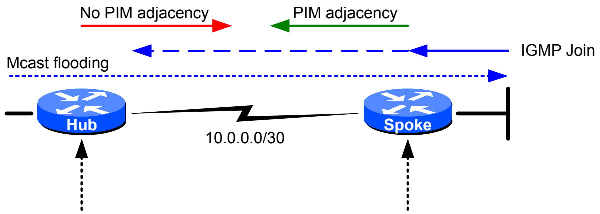
```
interface serial 0/0
 ip multicast helper-map 224.1.2.3 10.0.0.255 100

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
```

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

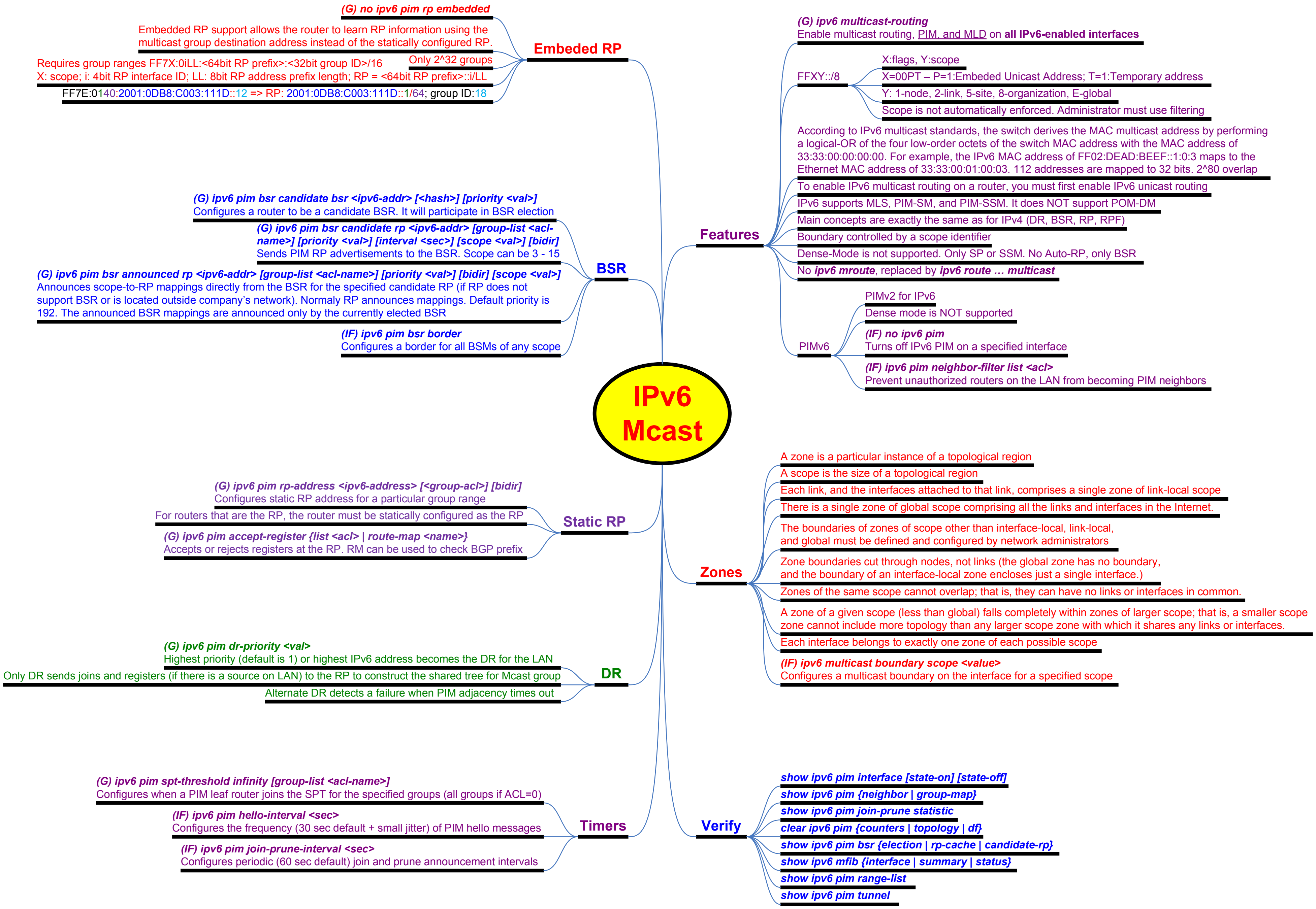
Multicast must be enabled on each interface, so mcast traffic can be flooded, but filtering must be used, so hub does not form PIM adjacency to spoke, so no automatic flooding is performed (in dense-mode)

(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
```



IPv6 Mcast

Embeded 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.
- Only 2^32 groups
- Requires group ranges FF7X:0iLL:<64bit RP prefix>:<32bit group ID>/16
  X: scope; i: 4bit RP interface ID; LL: 8bit RP address prefix length; RP = <64bit RP prefix>::i/LL
- FF7E:0140:2001:0DB8:C003:111D::12 => RP: 2001:0DB8:C003:111D::1/64; group ID:18

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
- ***(G) 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

Static RP

- ***(G) ipv6 pim rp-address <ipv6-address> [<group-acl>] [bidir]***
  Configures static RP address for a particular group range
- For routers that are the RP, the router must be statically configured as the RP
- ***(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

DR

- ***(G) 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

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

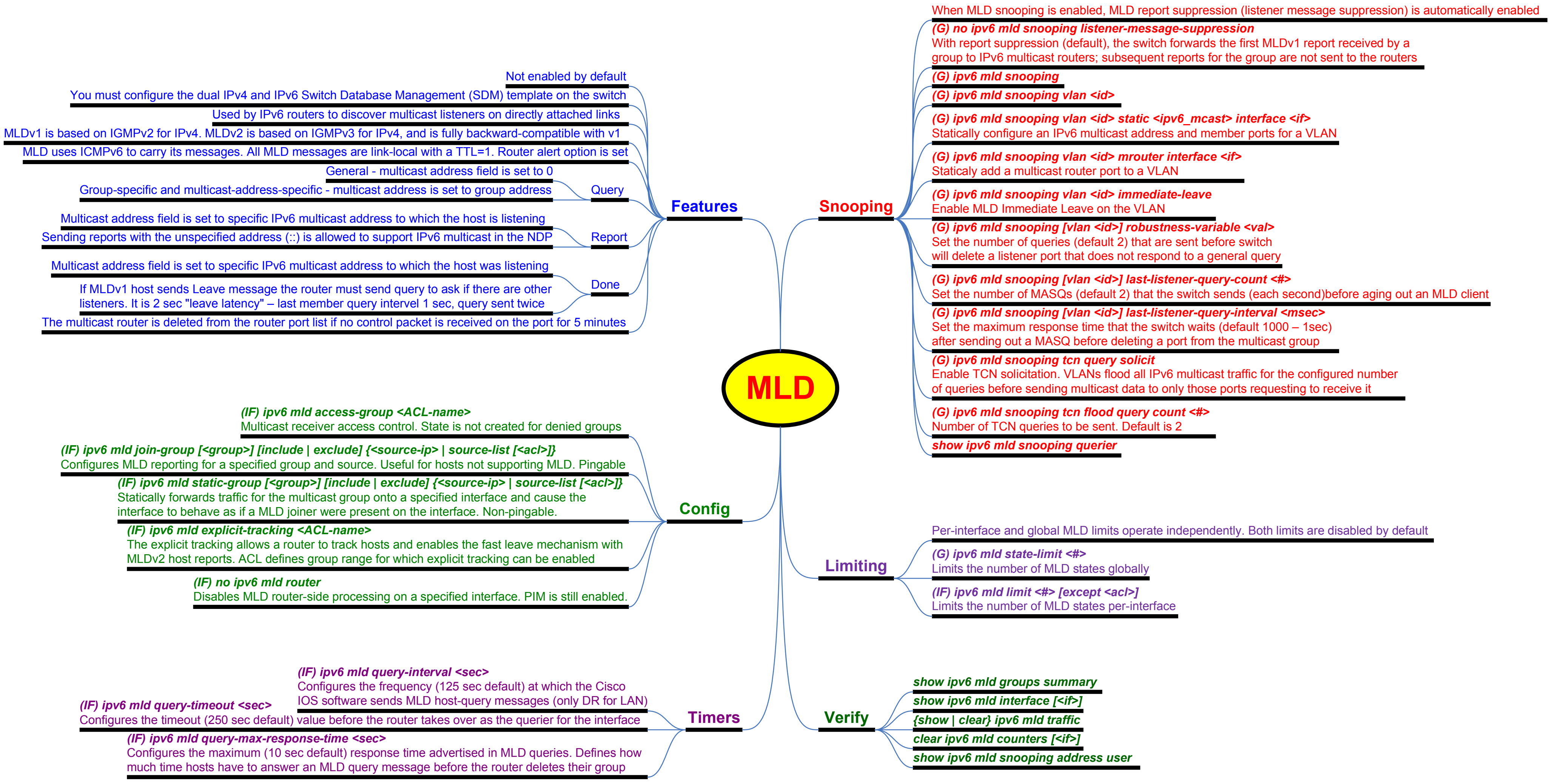
- ***(G) ipv6 multicast-routing***
  Enable multicast routing, PIM, and MLD on **all IPv6-enabled interfaces**
- FFXY::/8
  - X:flags, Y:scope
  - X=00PT – P=1:Embeded Unicast Address; T=1:Temporary address
  - Y: 1-node, 2-link, 5-site, 8-organization, E-global
  - Scope is not automatically enforced. Administrator must use filtering
- According to IPv6 multicast standards, the switch derives the MAC multicast address by performing a logical-OR of the four low-order octets of the switch MAC address with the MAC address of 33:33:00:00:00:00. For example, the IPv6 MAC address of FF02:DEAD:BEEF::1:0:3 maps to the Ethernet MAC address of 33:33:00:01:00:03. 112 addresses are mapped to 32 bits. 2^80 overlap
- 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
- Dense-Mode is not supported. Only SP or SSM. No Auto-RP, only BSR
- No ***ipv6 mroute***, replaced by ***ipv6 route … multicast***
- PIMv6
  - PIMv2 for IPv6
  - Dense mode is NOT supported
  - ***(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***
- ***show ipv6 pim tunnel***



MLD

Features

- Not enabled by default
- You must configure the dual IPv4 and IPv6 Switch Database Management (SDM) template on the switch
- 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
- The multicast router is deleted from the router port list if no control packet is received on the port for 5 minutes

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.

Timers

- ***(IF) 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)
- ***(IF) ipv6 mld query-timeout <sec>***
  Configures the timeout (250 sec default) value before the router takes over as the querier for the interface
- ***(IF) 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

Snooping

- When MLD snooping is enabled, MLD report suppression (listener message suppression) is automatically enabled
- ***(G) no ipv6 mld snooping listener-message-suppression***
  With report suppression (default), the switch forwards the first MLDv1 report received by a group to IPv6 multicast routers; subsequent reports for the group are not sent to the routers
- ***(G) ipv6 mld snooping***
- ***(G) ipv6 mld snooping vlan <id>***
- ***(G) ipv6 mld snooping vlan <id> static <ipv6_mcast> interface <if>***
  Statically configure an IPv6 multicast address and member ports for a VLAN
- ***(G) ipv6 mld snooping vlan <id> mrouter interface <if>***
  Staticaly add a multicast router port to a VLAN
- ***(G) ipv6 mld snooping vlan <id> immediate-leave***
  Enable MLD Immediate Leave on the VLAN
- ***(G) ipv6 mld snooping [vlan <id>] robustness-variable <val>***
  Set the number of queries (default 2) that are sent before switch will delete a listener port that does not respond to a general query
- ***(G) ipv6 mld snooping [vlan <id>] last-listener-query-count <#>***
  Set the number of MASQs (default 2) that the switch sends (each second)before aging out an MLD client
- ***(G) ipv6 mld snooping [vlan <id>] last-listener-query-interval <msec>***
  Set the maximum response time that the switch waits (default 1000 – 1sec) after sending out a MASQ before deleting a port from the multicast group
- ***(G) ipv6 mld snooping tcn query solicit***
  Enable TCN solicitation. VLANs flood all IPv6 multicast traffic for the configured number of queries before sending multicast data to only those ports requesting to receive it
- ***(G) ipv6 mld snooping tcn flood query count <#>***
  Number of TCN queries to be sent. Default is 2
- ***show ipv6 mld snooping querier***

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

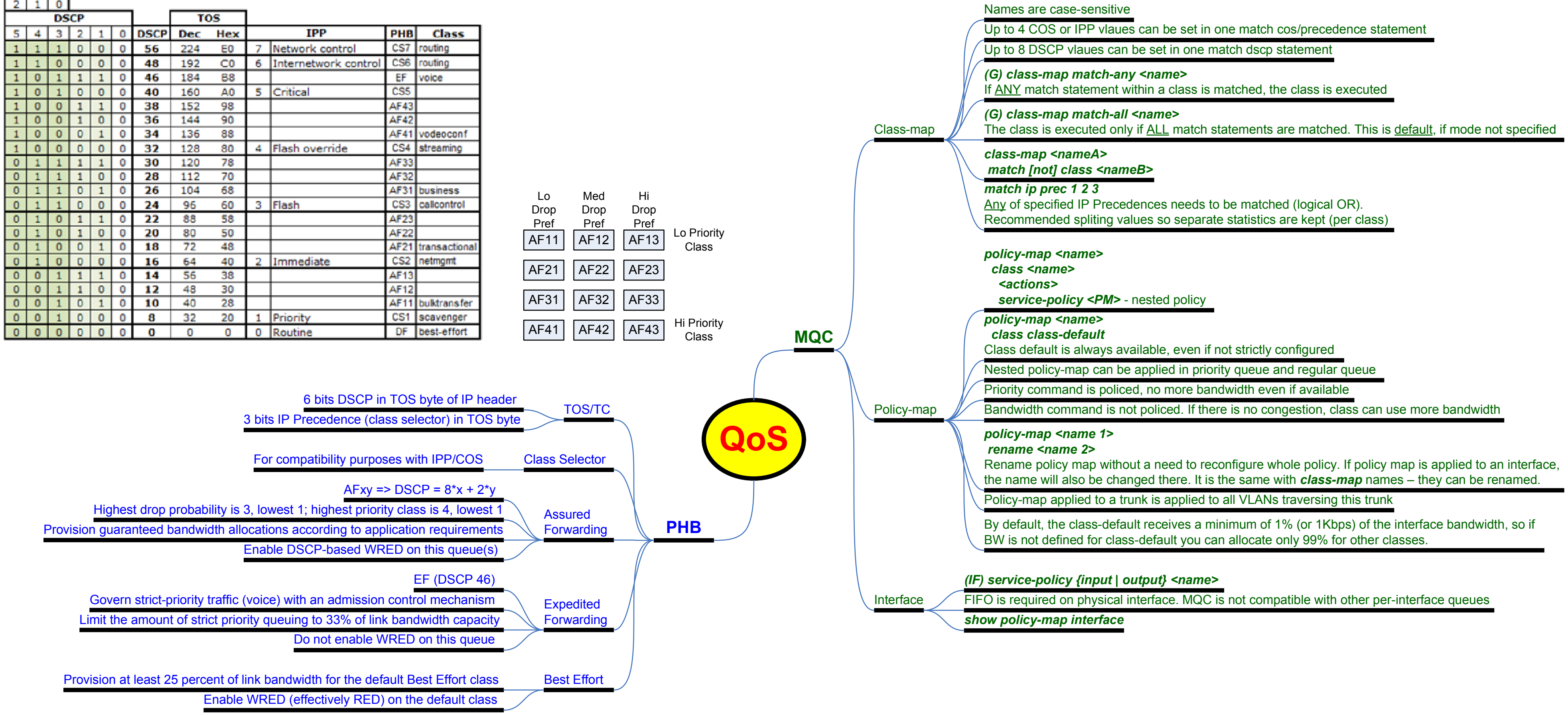
Verify

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

TOS							
7	6	5	4	3	2	1	0
IP Prec						ECN	
2	1	0					

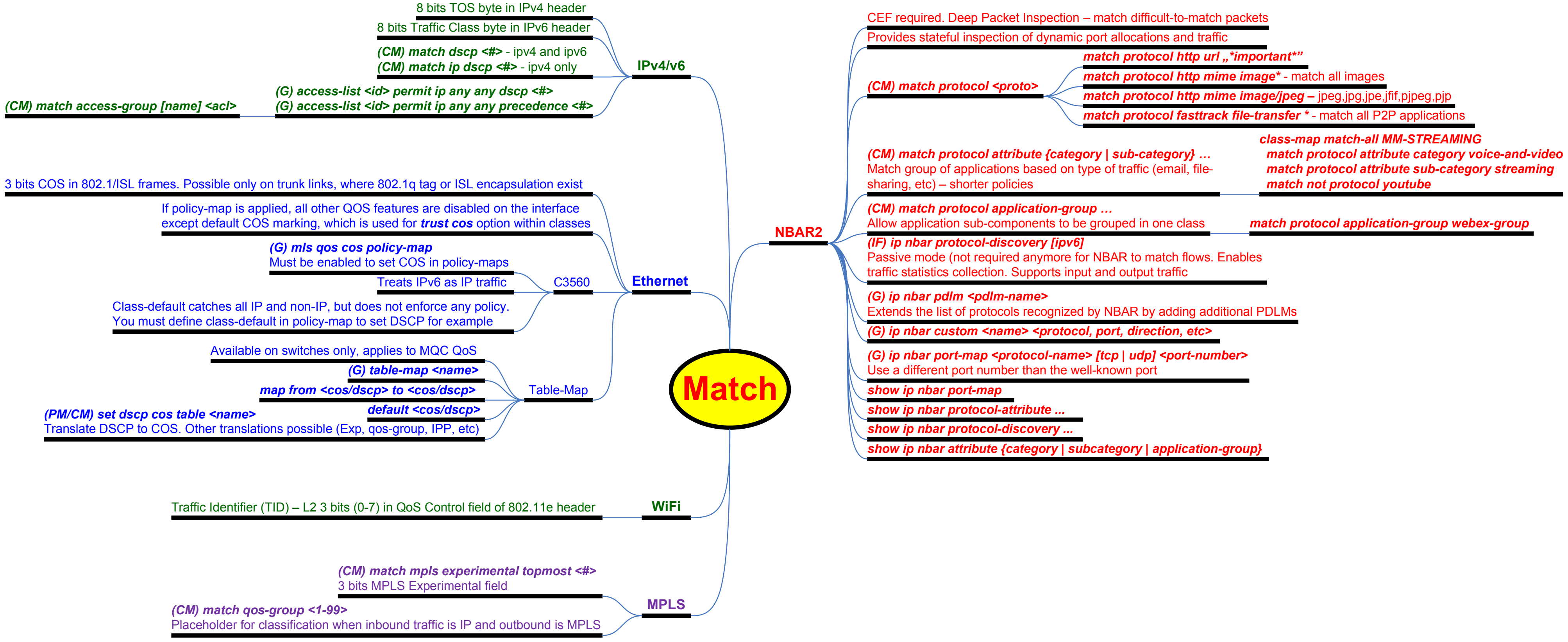
DSCP							TOS					
5	4	3	2	1	0	DSCP	Dec	Hex	IPP		PHB	Class
1	1	1	0	0	0	**56**	224	E0	7	Network control	CS7	routing
1	1	0	0	0	0	**48**	192	C0	6	Internetwork control	CS6	routing
1	0	1	1	1	0	**46**	184	B8			EF	voice
1	0	1	0	0	0	**40**	160	A0	5	Critical	CS5	
1	0	0	1	1	0	**38**	152	98			AF43	
1	0	0	1	0	0	**36**	144	90			AF42	
1	0	0	0	1	0	**34**	136	88			AF41	vodeoconf
1	0	0	0	0	0	**32**	128	80	4	Flash override	CS4	streaming
0	1	1	1	1	0	**30**	120	78			AF33	
0	1	1	1	0	0	**28**	112	70			AF32	
0	1	1	0	1	0	**26**	104	68			AF31	business
0	1	1	0	0	0	**24**	96	60	3	Flash	CS3	callcontrol
0	1	0	1	1	0	**22**	88	58			AF23	
0	1	0	1	0	0	**20**	80	50			AF22	
0	1	0	0	1	0	**18**	72	48			AF21	transactional
0	1	0	0	0	0	**16**	64	40	2	Immediate	CS2	netmgmt
0	0	1	1	1	0	**14**	56	38			AF13	
0	0	1	1	0	0	**12**	48	30			AF12	
0	0	1	0	1	0	**10**	40	28			AF11	bulktransfer
0	0	1	0	0	0	**8**	32	20	1	Priority	CS1	scavenger
0	0	0	0	0	0	**0**	0	0	0	Routine	DF	best-effort

Lo Drop Pref	Med Drop Pref	Hi Drop Pref	
AF11	AF12	AF13	Lo Priority Class
AF21	AF22	AF23	
AF31	AF32	AF33	
AF41	AF42	AF43	Hi Priority Class



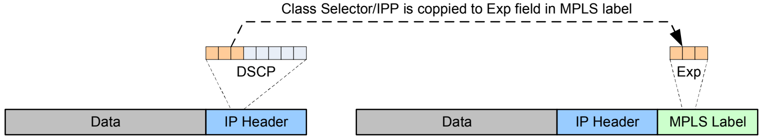
QoS

- **PHB**
  - TOS/TC
    - 6 bits DSCP in TOS byte of IP header
    - 3 bits IP Precedence (class selector) in TOS byte
  - Class Selector
    - For compatibility purposes with IPP/COS
  - Assured Forwarding
    - AFxy => DSCP = 8*x + 2*y
    - Highest drop probability is 3, lowest 1; highest priority class is 4, lowest 1
    - Provision guaranteed bandwidth allocations according to application requirements
    - Enable DSCP-based WRED on this queue(s)
  - Expedited Forwarding
    - EF (DSCP 46)
    - Govern strict-priority traffic (voice) with an admission control mechanism
    - Limit the amount of strict priority queuing to 33% of link bandwidth capacity
    - Do not enable WRED on this queue
  - Best Effort
    - Provision at least 25 percent of link bandwidth for the default Best Effort class
    - Enable WRED (effectively RED) on the default class
- **MQC**
  - Class-map
    - Names are case-sensitive
    - 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
    - ***(G) class-map match-any <name>***
      If ANY match statement within a class is matched, the class is executed
    - ***(G) class-map match-all <name>***
      The class is executed only if ALL match statements are matched. This is default, if mode not specified
    - ***class-map <nameA>***
      match [not] class <nameB>
    - ***match ip prec 1 2 3***
      Any of specified IP Precedences needs to be matched (logical OR).
      Recommended spliting values so separate statistics are kept (per class)
  - Policy-map
    - ***policy-map <name>***
      class <name>
      <actions>
      service-policy <PM> - nested policy
    - ***policy-map <name>***
      class class-default
      Class default is always available, even if not strictly configured
    - Nested policy-map can be applied in priority queue and regular queue
    - Priority command is policed, no more bandwidth even if available
    - Bandwidth command is not policed. If there is no congestion, class can use more bandwidth
    - ***policy-map <name 1>***
      rename <name 2>
      Rename policy map without a need to reconfigure whole policy. If policy map is applied to an interface, the name will also be changed there. It is the same with ***class-map*** names – they can be renamed.
    - Policy-map applied to a trunk is applied to all VLANs traversing this trunk
    - By default, the class-default receives a minimum of 1% (or 1Kbps) of the interface bandwidth, so if BW is not defined for class-default you can allocate only 99% for other classes.
  - Interface
    - ***(IF) service-policy {input | output} <name>***
    - FIFO is required on physical interface. MQC is not compatible with other per-interface queues
    - ***show policy-map interface***



Match

- **IPv4/v6**
  - 8 bits TOS byte in IPv4 header
  - 8 bits Traffic Class byte in IPv6 header
  - ***(CM) match dscp <#>*** - ipv4 and ipv6
(CM) match ip dscp <#> - ipv4 only
  - ***(G) access-list <id> permit ip any any dscp <#>***
(G) access-list <id> permit ip any any precedence <#>
    - ***(CM) match access-group [name] <acl>***
- **Ethernet**
  - 3 bits COS in 802.1/ISL frames. Possible only on trunk links, where 802.1q tag or ISL encapsulation exist
  - If policy-map is applied, all other QOS features are disabled on the interface except default COS marking, which is used for ***trust cos*** option within classes
  - C3560
    - ***(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, but does not enforce any policy. You must define class-default in policy-map to set DSCP for example
  - Table-Map
    - Available on switches only, applies to MQC QoS
    - ***(G) table-map <name>***
    - ***map from <cos/dscp> to <cos/dscp>***
    - ***default <cos/dscp>***
    - ***(PM/CM) set dscp cos table <name>***
Translate DSCP to COS. Other translations possible (Exp, qos-group, IPP, etc)
- **WiFi**
  - Traffic Identifier (TID) – L2 3 bits (0-7) in QoS Control field of 802.11e header
- **MPLS**
  - ***(CM) match mpls experimental topmost <#>***
3 bits MPLS Experimental field
  - ***(CM) match qos-group <1-99>***
Placeholder for classification when inbound traffic is IP and outbound is MPLS
- **NBAR2**
  - CEF required. Deep Packet Inspection – match difficult-to-match packets
  - Provides stateful inspection of dynamic port allocations and traffic
  - ***(CM) match protocol <proto>***
    - ***match protocol http url „*important*”***
    - ***match protocol http mime image**** - match all images
    - ***match protocol http mime image/jpeg –*** jpeg,jpg,jpe,jfif,pjpeg,pjp
    - ***match protocol fasttrack file-transfer **** - match all P2P applications
  - ***(CM) match protocol attribute {category | sub-category} …***
Match group of applications based on type of traffic (email, file-sharing, etc) – shorter policies
    - ***class-map match-all MM-STREAMING***
match protocol attribute category voice-and-video
match protocol attribute sub-category streaming
match not protocol youtube
  - ***(CM) match protocol application-group …***
Allow application sub-components to be grouped in one class
    - ***match protocol application-group webex-group***
  - ***(IF) ip nbar protocol-discovery [ipv6]***
Passive mode (not required anymore for NBAR to match flows. Enables traffic statistics collection. Supports input and output traffic
  - ***(G) ip nbar pdlm <pdlm-name>***
Extends the list of protocols recognized by NBAR by adding additional PDLMs
  - ***(G) ip nbar custom <name> <protocol, port, direction, etc>***
  - ***(G) ip nbar port-map <protocol-name> [tcp | udp] <port-number>***
Use a different port number than the well-known port
  - ***show ip nbar port-map***
  - ***show ip nbar protocol-attribute …***
  - ***show ip nbar protocol-discovery …***
  - ***show ip nbar attribute {category | subcategory | application-group}***



Queue

WFQ in MQC

- HQF – Hierarchical Queueing Framework aka. CBWFQ
- 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)
  - ***(CM) 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.
  - ***(CM) fair-queue [<# of dynamic conv>]***
    In class-default only <12.4.20T. All classes in later IOS

PQ/LLQ

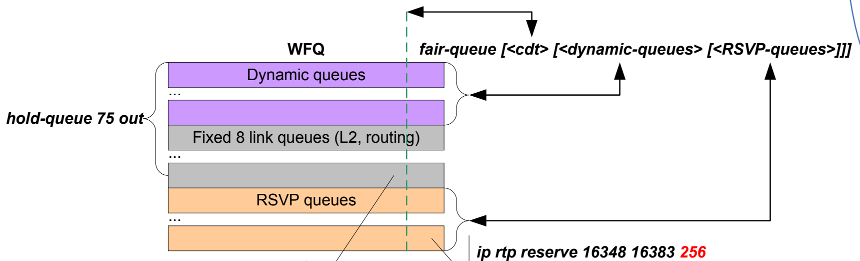
- Policies traffic up to defined priority BW
- BW + PQ is still limited to 75% of intf BW
- ***(CM) priority {<bw> | percent <%>} [<burst>]***
  Burst by default 200ms of traffic. May be adjusted for video applicaitions (Ex.: 64kB in 33ms frame)
- Unlike bandwidth, priority can use percent and remaining-percent 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
- ***(IF) fair-queue [<cdt> [<dynamic-queues> [<RSVP-queues>]]]***
- Once traffic is emptied from one flow queue, the flow queue is removed, even if TCP session between two hosts is still up
- Modified tail drop
  - CDT – Congestion avoidance scheme available in WFQ. When CDT threshold is reached WFQ drops packet from a flow queue with max virtual scheduling time.
  - 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
fair-queue [<cdt> [<dynamic-queues> [<RSVP-queues>]]]
Dynamic queues
...
Fixed 8 link queues (L2, routing)
...
RSVP queues
...
hold-queue 75 out

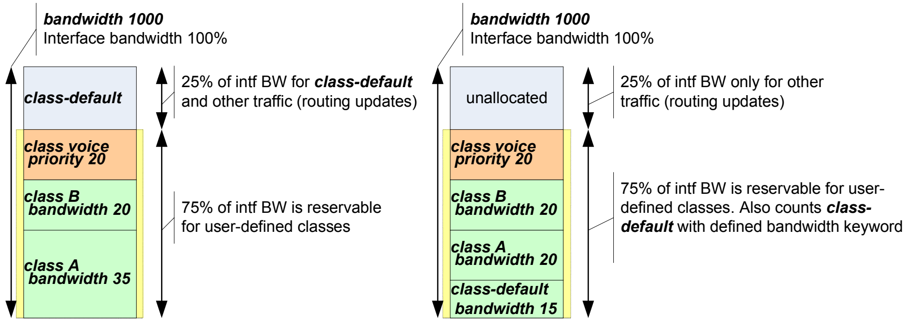
ip rtp priority 16348 16383 256
This queue gets weight 0 and is policed up to 256k. Also, only even UDP ports are considered. Voice always gets priority. This queue sits just right after 8 link queues

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

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 <%>*** **Deprecated!**
  - If class-default has bandwidth defined it is also calculated as reservable

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



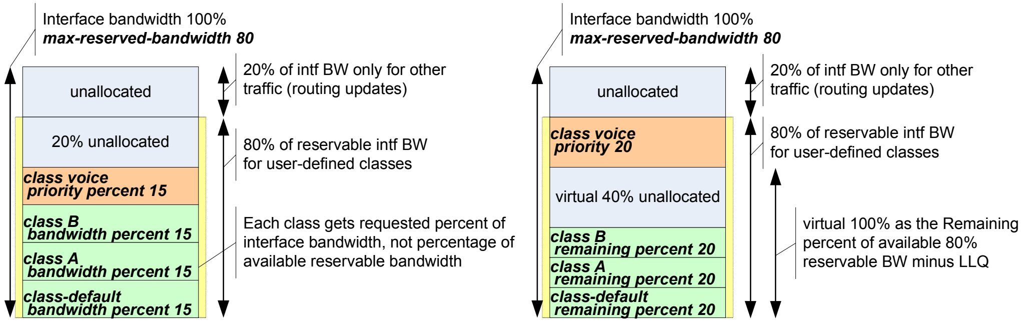
25% of intf BW for ***class-default*** and other traffic (routing updates)

75% of intf BW is reservable for user-defined classes

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
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

Each class gets requested percent of interface bandwidth, not percentage of available reservable bandwidth

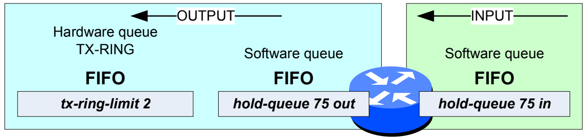
20% of intf BW only for other traffic (routing updates)

80% of reservable intf BW for user-defined classes

virtual 100% as the Remaining percent of available 80% reservable BW minus LLQ

TX-Ring

- 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 on TX-ring
- ***(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 (default, not changed by different queueing or manual setting). 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 queue is always FIFO (default 75 packets)
- ***(IF) hold-queue <#> {in | out}***
- ***(CM) no fair-queue***
  Enable FIFO on the class



Policing

Concept

Can be applied inbound and aoubound, but usually used as inbound conformation of the allowed traffic (the ISP polices inbound traffic, and the customer shapes his outgoing traffic)

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*

Policing counts TCP/IP headers

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

Multiaction (remarking, dropping)

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

(CM) police <cir> <burst> exceed-action policed-dscp-transmit
Remarking of exceeding traffic using policed-dscp map

(CM) police <cir> <burst> exceed-action drop
Policing can be set for ingress policy-map per interface

Abbr (k, m, g) can be used for speed (ex.: 10.5m)

CAR

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

(IF) 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 (125 ms) as Burst is in Bytes. Bc = (CIR/8)*(Tc/1000)

Statements 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

(IF) 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

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

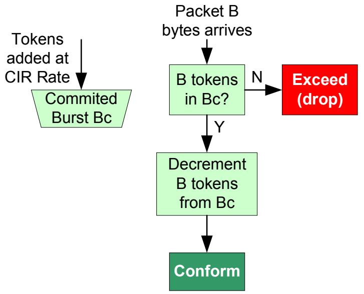
Number of bits in packet is compared to number of available tokens in a bucket. Packet is either transmited or dropped.

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

Allows bursts as long as overall average is below CIR

Variation of cumulated tokens is unpredictible

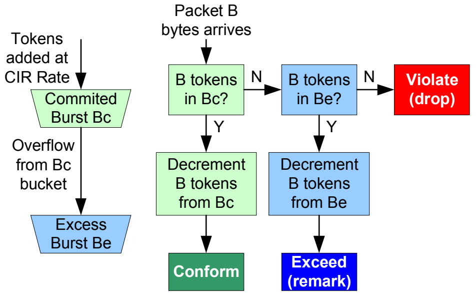
Two buckets; Three actions: Conform, Exceed, Violate

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



Two-rate Three-color

Unpredictability from one-rate 3-color fixed with PIR rate

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

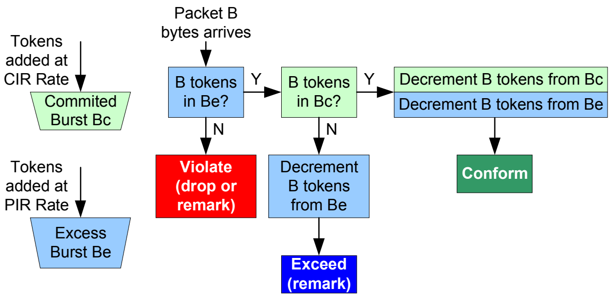
Be is filled twice faster that Bc. If Bc (CIR) = 128, then Be (PIR) = 256k. During conform action tokets are taken from both buckets

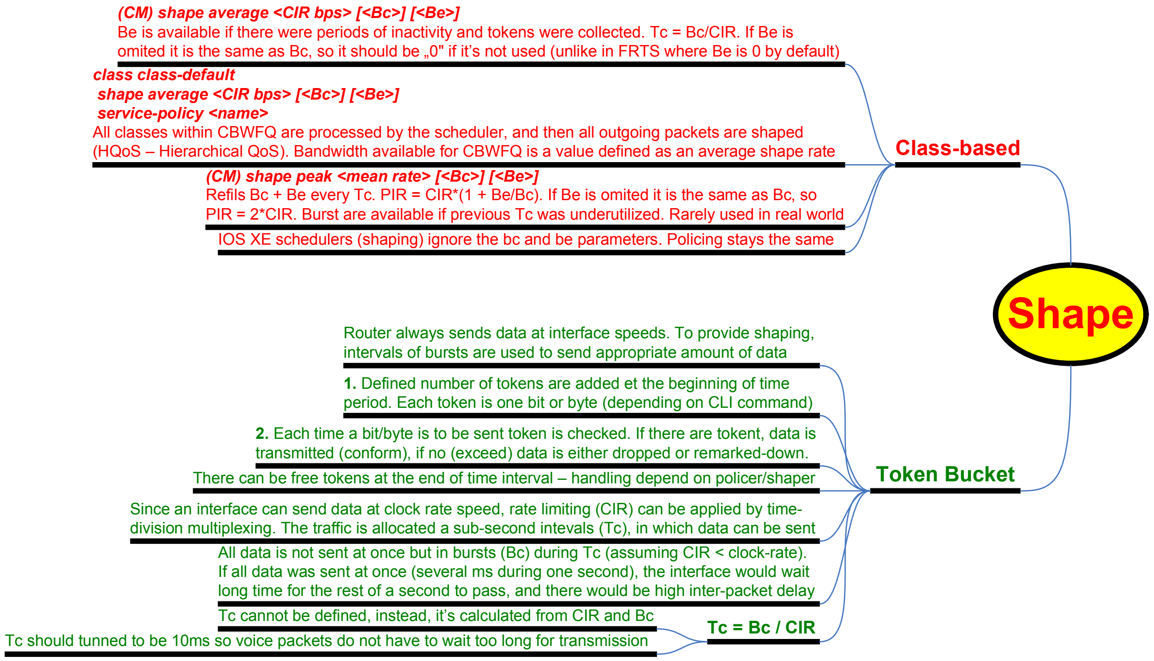
police cir <cir> [bc <Bc>] pir <pir> [be <Be>] conform-action ...

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





Shape

Class-based

(CM) 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 (HQoS – Hierarchical QoS). Bandwidth available for CBWFQ is a value defined as an average shape rate

(CM) 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. Burst are available if previous Tc was underutilized. Rarely used in real world

IOS XE schedulers (shaping) ignore the bc and be parameters. Policing stays the same

Token Bucket

Router always sends data at interface speeds. To provide shaping, intervals of bursts are used to send appropriate amount of data

1. Defined number of tokens are added et the beginning of time period. Each token is one bit or byte (depending on CLI command)

2. Each time a bit/byte is to be sent token is checked. If there are tokent, data is transmitted (conform), if no (exceed) data is either dropped or remarked-down.

There can be free tokens at the end of time interval – handling depend on policer/shaper

Since an interface can send data at clock rate speed, rate limiting (CIR) can be applied by time-division multiplexing. The traffic is allocated a sub-second intevals (Tc), in which data can be sent

All data is not sent at once but in bursts (Bc) during Tc (assuming CIR < clock-rate). If all data was sent at once (several ms during one second), the interface would wait long time for the rest of a second to pass, and there would be high inter-packet delay

Tc = Bc / CIR

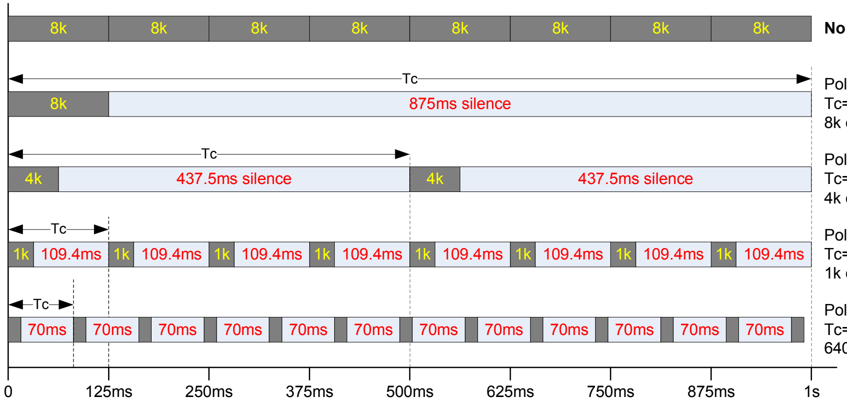
Tc cannot be defined, instead, it's calculated from CIR and Bc

Tc should tunned to be 10ms so voice packets do not have to wait too long for transmission

Ex.: CIR set to 8000bps on 64000bps link, data 8000b to be sent

Data sent: 8k data / 64k clock = 125ms <= only during this time sending is allowed

8k Data sent at 64kb/s 10ms Silence



No shaping/policing, 64k line rate speed

Policing 8kbps. Bc set to 8000 <= for policed data
Tc=8000/8000=1sec <= 1 interval in 1 sec
8k on 64k link takes 125ms to transmit

Policing 8kbps. Bc set to 4000 <= for policed data
Tc=4000/8000=500ms <= 2 intervals in 1 sec
4k on 64k link takes 62,5ms to transmit

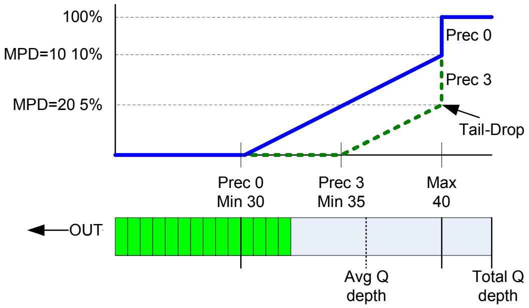
Policing 8kbps. Bc set to 1000 <= for policed data
Tc=1000/8000=125ms <= 8 intervals in 1 sec
1k on 64k link takes 15,62ms to transmit

Policing 8kbps. Bc set to 640 <= for policed data
Tc=640/8000=80ms <= 12,5 intervals in 1 sec
640b on 64k link takes **10ms** to transmit

WRED

Features

- Enable DSCP-based WRED on AF and DF queues. Do not use WRED on EF and controll traffic. Scavenger also does not requre WRED.
- Tail-drop causes global synchronization (slow-start) and saw-shaped traffic graph
- TCP Starvation – mixing TCP and UDP traffic in the same class, and controling congestion for TCP makes more room for UDP



ECN

- ***(G) ip tcp ecn***
  Enable TCP Explicit Congestion Notification
- 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

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
- ***(CM) 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.

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>***

L2 QoS

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.

Switch

- Existing QoS configurations are overriden when Auto Qos is configured on port
- ***(IF) auto qos voip trust***
  The switch trusts CoS for switched ports or DSCP for routed ports. Adds „mls qos trust cos/dscp" to the interface. Unconditional trust
- ***(IF) auto qos voip cisco-phone***
  Conditional trust. If IP Phone is detected using CDP then port trusts CoS. If phone is not present all marking is reset to 0. Ingress and egress queues are configured. Adds „mls qos trust cos" to the interface. Adds „mls qos trust cos" to the interface
- ***(IF) auto qos voip cisco-softphone***
  Switch applies policy-map to the interface with classification and marking
- ***(IF) auto qos classify [police]***
  configure the QoS for trusted interfaces. Detailed policy-map with classes and ACLs is created and applied to the interface. Either plain marking with DSCP or in addition with policing each class
- ***(IF) auto qos trust [{cos | dscp}]***
  Unconditional trust. Adds „mls qos trust cos/dscp". If classification is ommited, then COS is used as default (even on L3 port)

Switch port trust state

- ***(IF) mls qos trust dscp***
  If switch trusts 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. Recommended trust state due to high granularity
- ***(IF) 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 (watch for default mapping for COS5)
- ***(IF) mls qos trust device cisco-phone***
  Conditional trust. Enabled when switch detects IP Phone using CDPv2. Trust COS must be used on that port
- ***(IF) qos trust device cisco-phone***
  Trust configuration on 4500
- ***(IF) trust device {cisco-phone | cts | ip-camera | media-player}***
  Trust configuration on 3650/3850
- ***(IF) switchport priority extend [cos <cos> | trust]***
  Used in conjunction with ***mls qos trust device cisco-phone.*** Overwrites the original CoS value of all Ethernet frames received from PC attached to IP phone with the value specified (COS=0 is default). IP Phone is unable to mark DSCP
- ***(IF) 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.
- ***(IF) 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.
  - ***(IF) no mls qos rewrite ip dscp***
    Cat 3560. Does not change DSCP in the packet. Use mapping to derive internal DSCP, but DSCP in the packet is not changed.
- ***show mls qos interface***
- ***show mls qos map***

Maps

Non-IP Traffic

- Trust the CoS value in the incoming frame (configure the port to trust CoS). Then use the configurable CoS-to-DSCP map to generate a DSCP value for the packet
- Trust the DSCP or trust IP precedence configurations are meaningless for non-IP traffic. If you configure a port with either of these options and non-IP traffic is received, the switch assigns a CoS value and generates an internal DSCP value from the CoS-to-DSCP map. The switch uses the internal DSCP value to generate a CoS value representing the priority of the traffic

IP Traffic

- Trust the DSCP value in the incoming packet (configure the port to trust DSCP), and assign the same DSCP value to the packet. For ports that are on the boundary between two QoS administrative domains, you can modify the DSCP to another value by using the configurable DSCP-to-DSCP-mutation map
- Trust the CoS value (if present) in the incoming packet, and generate a DSCP value for the packet by using the CoS-to-DSCP map. If the CoS value is not present, use the default port CoS value
- Override the configured CoS of incoming packets, and apply the default port CoS value to them. For IPv6 packets, the DSCP value is rewritten by using the CoS-to-DSCP map and by using the default CoS of the port. You can do this for both IPv4 and IPv6 traffic

- 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

CoS-to-DSCP

- ***(G) mls qos map cos-dscp <dscp1>...<dscp8>***
  Default map: 0 8 16 24 32 40 48 56. VoIP falls under 40, so COS5 should be changed to 46 (EF)
- Map CoS values in incoming packets to a DSCP value that QoS uses internally to represent the priority of the traffic

IPPrec-to-DSCP

- ***(G) 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

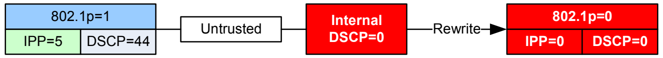
- The default policed-DSCP map is a null map, which maps an incoming DSCP value to the same DSCP value
- ***(G) mls qos map policed-dscp <dscp1>...<dscp8> to <mark-down-dscp>***
  Mark down a DSCP value to a new value as the result of a policing and marking action

DSCP-to-CoS

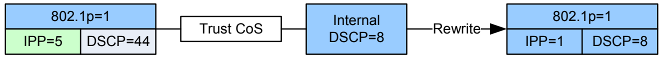
- ***(G) mls qos map dscp-cos <dscp1>...<dscp8> to <cos>***
  Generate a CoS value, which is used to select one of the four egress queues

DSCP-to-DSCP Mutation

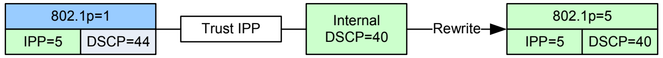
- If the two domains have different DSCP definitions between them, use the DSCP-to-DSCP-mutation map to translate a set of DSCP values to match the definition of the other domain
- Original map cannot be changed, you can manipulate a copy and assign it to specific interface. The other option is CBWFQ 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>



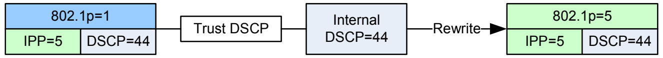
When port is untrusted, internal DSCP is 0, and all values are reset to 0 on outgoing intf



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

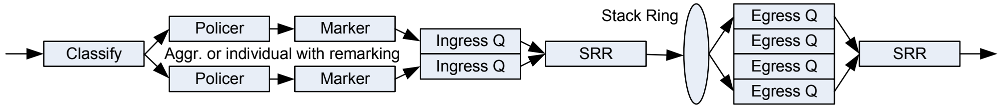


When port trusts IPP, internal DSCP is taken from IPP-to-DSCP mapping. Outgoing interface rewrites DSCP and CoS accordingly to internal DSCP.



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

Catalyst 2960 / 3560 / 3750 are the last platforms to use *mls qos* syntax



3560 QoS

Features

(G) mls qos
QoS is disabled by default. Packets are not modified (CoS, DSCP, and IPP in the packet are not changed). When enabled all ports become untrusted (set COS 0)

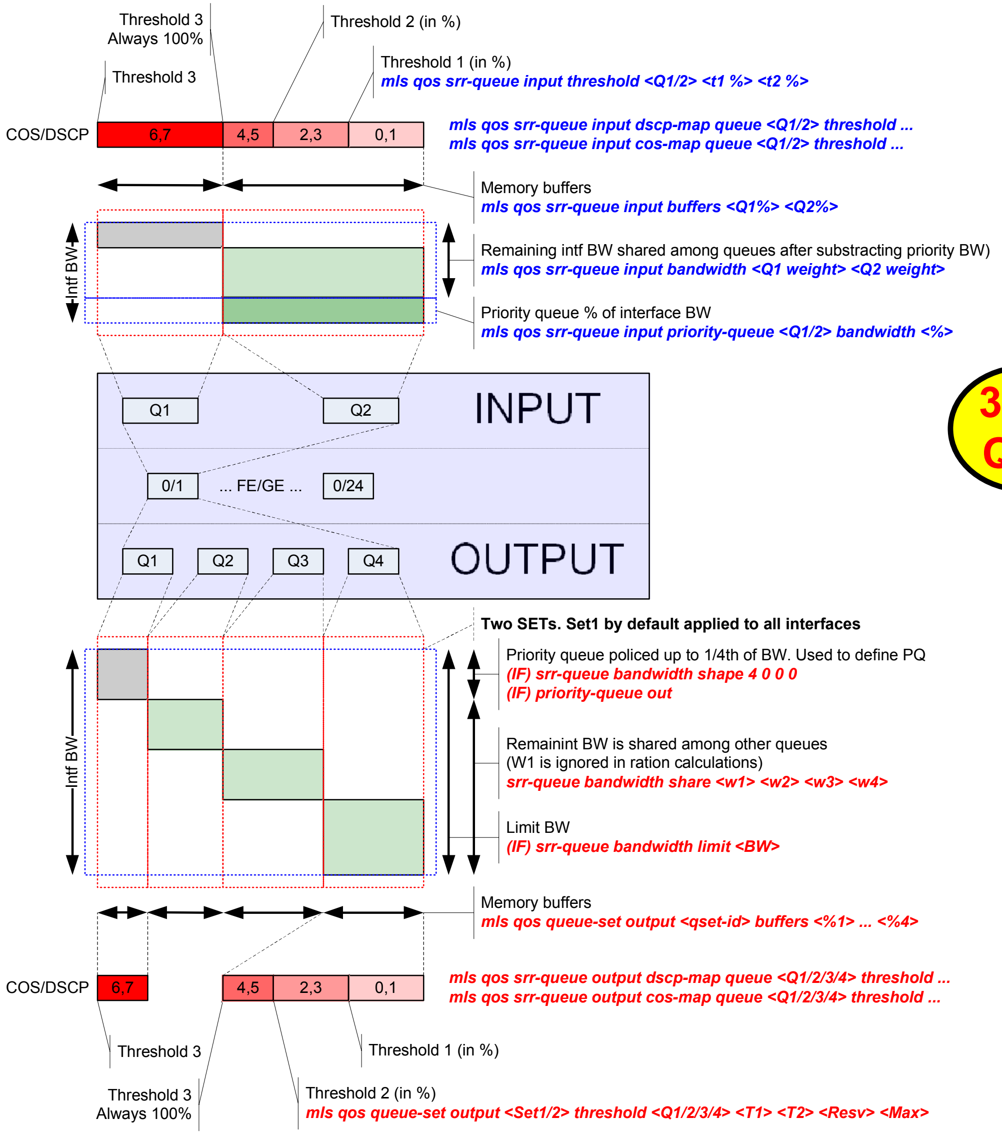
When using port-channel, QoS must be enabled on physical links

Control traffic (BPDU, routing) are subject to ingress QoS

VLAN based

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

(SVI) service-policy input <name>
This policy will be inherited by ports using those VLANs in access mode



mls qos srr-queue input threshold <Q1/2> <t1 %> <t2 %>

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%>

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 <%>

Two SETs. Set1 by default applied to all interfaces

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>

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 ...

mls qos queue-set output <Set1/2> threshold <Q1/2/3/4> <T1> <T2> <Resv> <Max>

Ingress Queue 1P1Q3T (2Q3T)

The switch supports two configurable ingress queues, which are serviced by SRR in shared mode only (with WTD)

Scheduler - Shaped Round Robin with sharing method as the only supported mode for ingress

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-editable)
(IF) 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
(IF) mls qos srr-queue input dscp-map queue <Q1/2> threshold <T1/2/3> <dscp1-8>
(IF) 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
(IF) 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 Q2 can have priority
mls qos srr-queue input priority-queue <Q1/2> bandwidth <% of interface>

show mls qos input-queue

show mls qos maps {cos-input-q | dscp-input-q}

1P1Q3T	
EF CS5 CS4	P2
CS7 CS6	Q1T3
CS3	Q1T2
AF4 AF3 AF2 CS2 AF1 CS1 DF	Q1T1

Egress queue 1P3Q3T (4Q3T)

Shaped Round Robin (SRR) with Weighted Tail Drop

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, even if other queues are empty. Weights are in inverse: 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 is shaped according to weights defined in ***share*** command. Defines PQ (***priority-queue out*** must be used on interface)

Shared

Ratio of the weights controls the frequency of dequeuing; the absolute values are meaningless

(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. If one port has empty resources (nothing is plugged in) they can be used by other ports. Reserved: port gets on start; Max: if needed, up to this % 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
(IF) mls qos srr-queue output dscp-map queue <Q1/2/3/4> threshold <T1/2/3> <dscp1-8>
(IF) 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%
(IF) mls qos queue-set output <qset-id> buffers <%1> ... <%4>

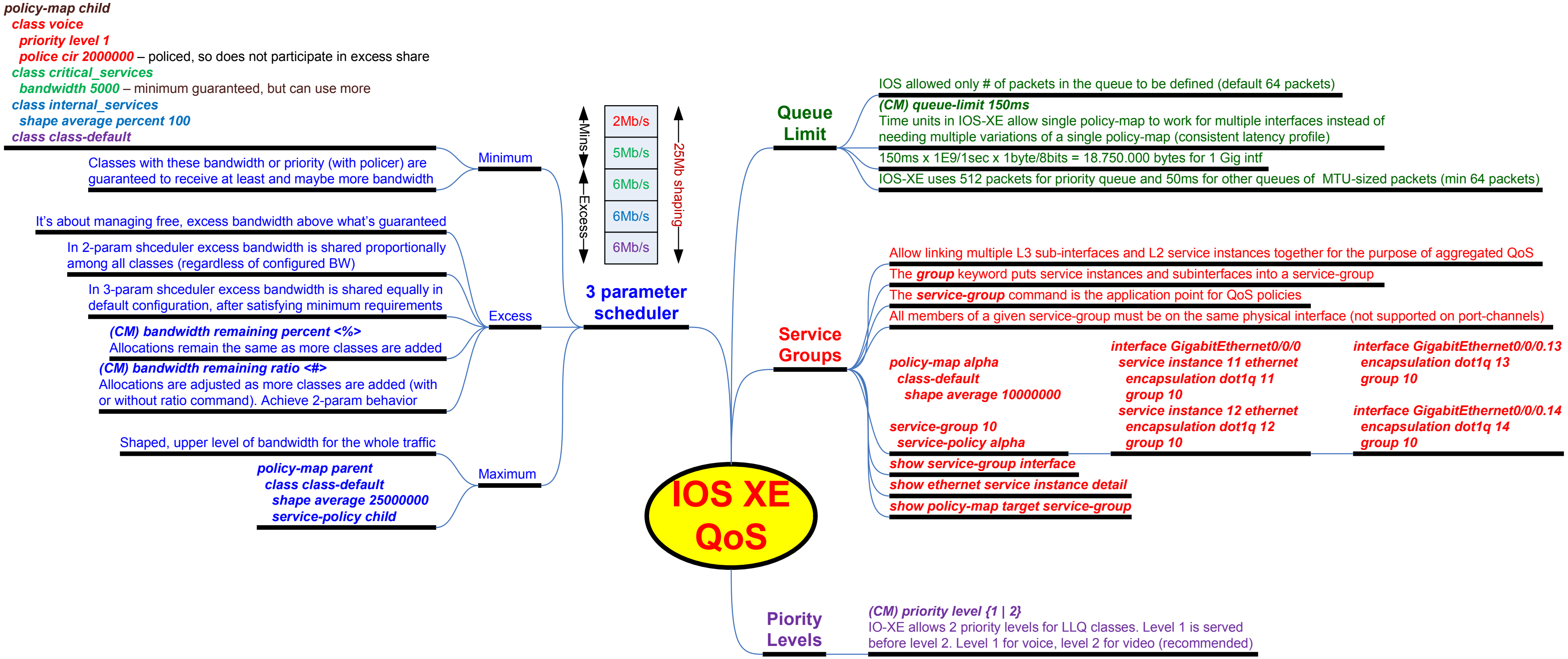
4. Limit bandwidth
Configurable 10-90% of physical BW on 6Mb basis. If you define 10, the limit will be 6-12Mb
(IF) srr-queue bandwidth limit <BW>

(IF) queue-set {1 | 2}
Assign queue set to an interface. Set 1 is alredy assigned to all ports, so use only if you apply set 2

show mls qos interface <IF> {queueing | statistics}

show mls qos queue-set {1 | 2}

1P1Q3T	
CS1	Q4T2
AF1	Q4T1
DF	Q3
CS6 CS7	Q2T3
CS3	Q2T2
AF4 AF3 AF2 CS2	Q2T1
EF CS5 CS4	P1



IOS XE QoS

3 parameter scheduler

Minimum

```
policy-map child
  class voice
    priority level 1
    police cir 2000000 – policed, so does not participate in excess share
  class critical_services
    bandwidth 5000 – minimum guaranteed, but can use more
  class internal_services
    shape average percent 100
  class class-default
```

Classes with these bandwidth or priority (with policer) are guaranteed to receive at least and maybe more bandwidth

Excess

It's about managing free, excess bandwidth above what's guaranteed

In 2-param shceduler excess bandwidth is shared proportionally among all classes (regardless of configured BW)

In 3-param shceduler excess bandwidth is shared equally in default configuration, after satisfying minimum requirements

(CM) bandwidth remaining percent <%>
Allocations remain the same as more classes are added

(CM) bandwidth remaining ratio <#>
Allocations are adjusted as more classes are added (with or without ratio command). Achieve 2-param behavior

Maximum

Shaped, upper level of bandwidth for the whole traffic

```
policy-map parent
  class class-default
    shape average 25000000
    service-policy child
```

Queue Limit

IOS allowed only # of packets in the queue to be defined (default 64 packets)

(CM) queue-limit 150ms
Time units in IOS-XE allow single policy-map to work for multiple interfaces instead of needing multiple variations of a single policy-map (consistent latency profile)

150ms x 1E9/1sec x 1byte/8bits = 18.750.000 bytes for 1 Gig intf

IOS-XE uses 512 packets for priority queue and 50ms for other queues of MTU-sized packets (min 64 packets)

Service Groups

Allow linking multiple L3 sub-interfaces and L2 service instances together for the purpose of aggregated QoS

The ***group*** keyword puts service instances and subinterfaces into a service-group

The ***service-group*** command is the application point for QoS policies

All members of a given service-group must be on the same physical interface (not supported on port-channels)

```
policy-map alpha
  class-default
    shape average 10000000

service-group 10
  service-policy alpha
```

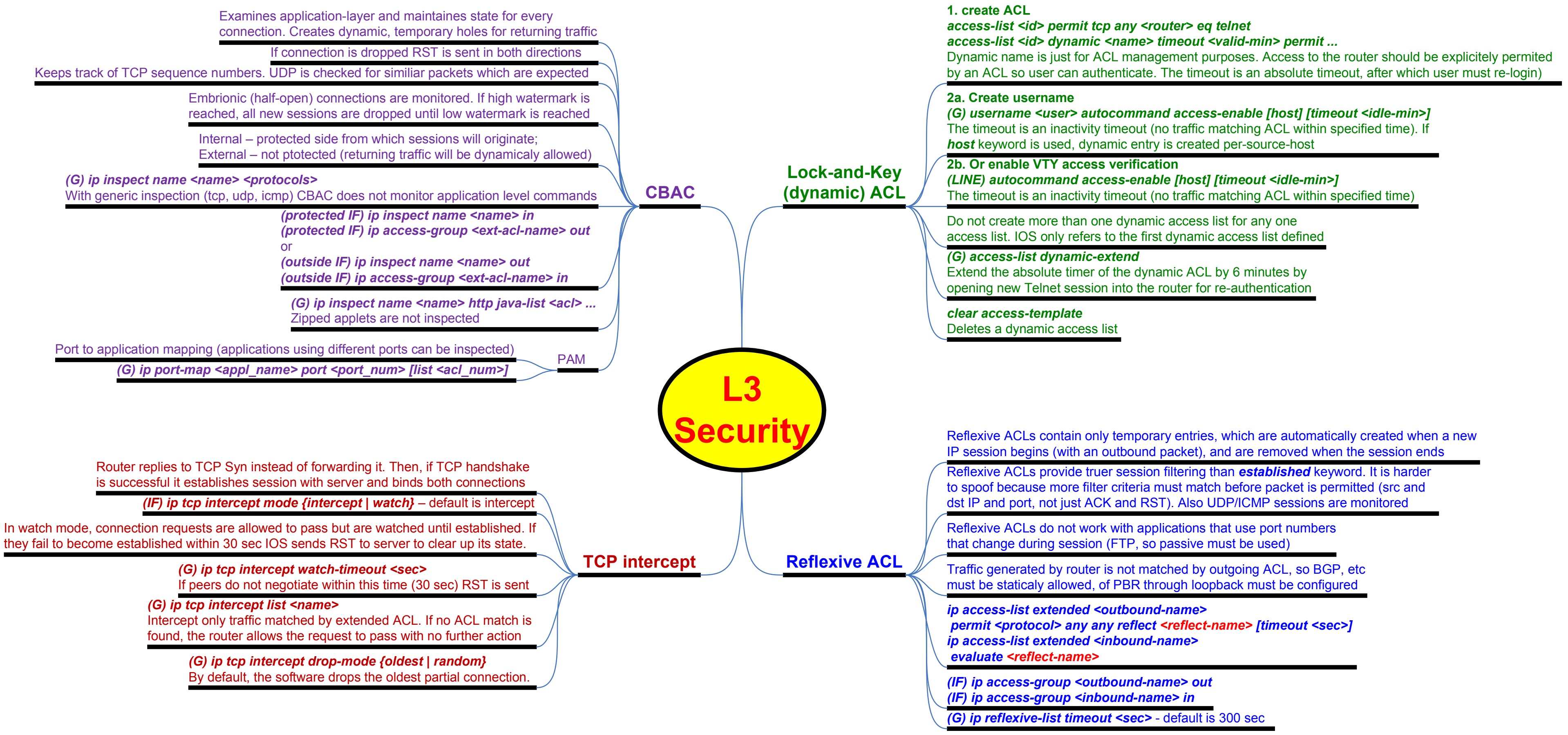
```
interface GigabitEthernet0/0/0
  service instance 11 ethernet
    encapsulation dot1q 11
    group 10
  service instance 12 ethernet
    encapsulation dot1q 12
    group 10
```

```
interface GigabitEthernet0/0/0.13
  encapsulation dot1q 13
  group 10

interface GigabitEthernet0/0/0.14
  encapsulation dot1q 14
  group 10
```

```
show service-group interface
show ethernet service instance detail
show policy-map target service-group
```

Piority Levels

(CM) priority level {1 | 2}
IO-XE allows 2 priority levels for LLQ classes. Level 1 is served before level 2. Level 1 for voice, level 2 for video (recommended)



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)
- ***(G) 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
- ***(G) 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)
  - ***(G) ip port-map <appl_name> port <port_num> [list <acl_num>]***

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

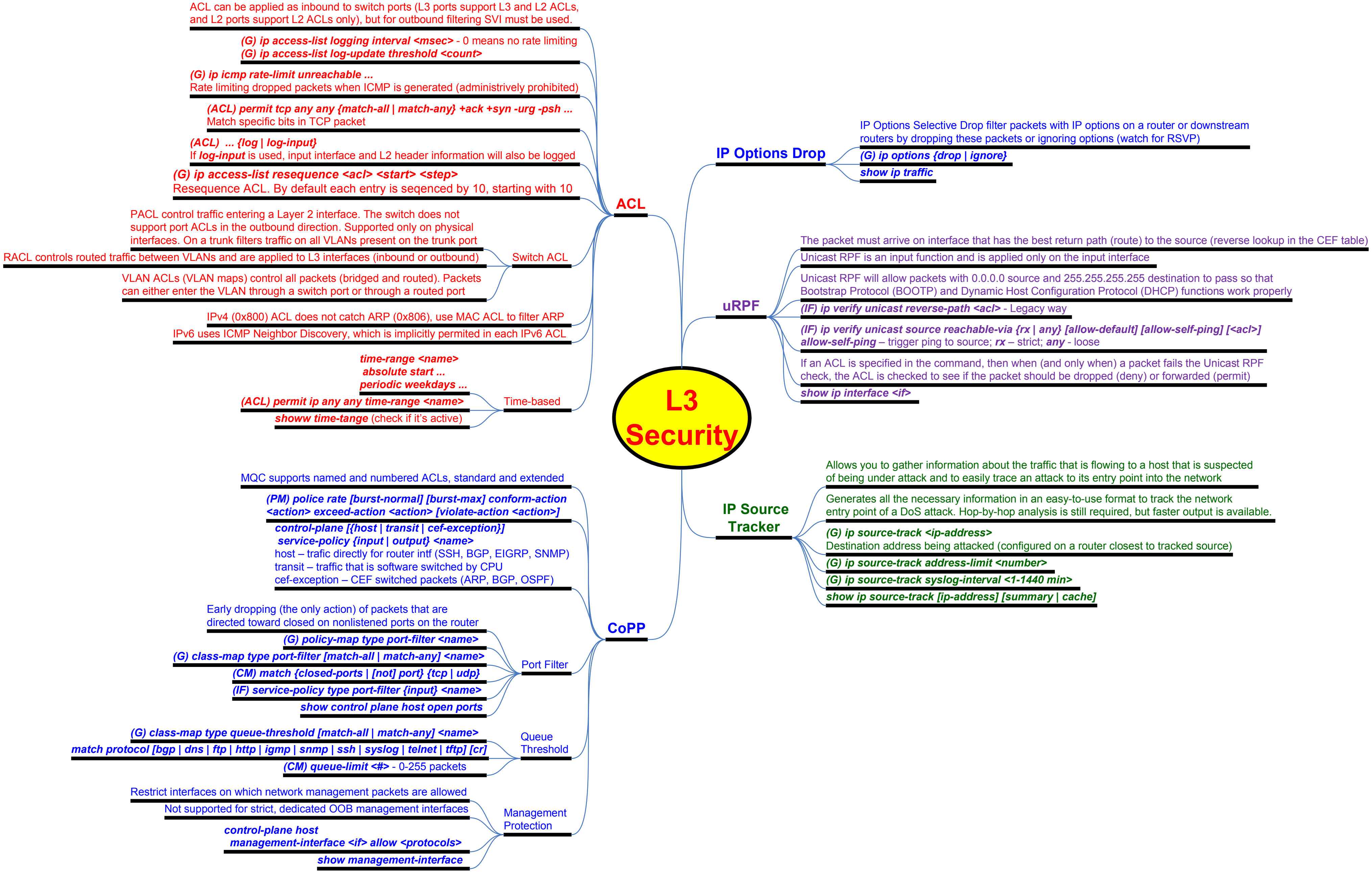
TCP intercept

- Router replies to TCP Syn instead of forwarding it. Then, if TCP handshake is successful it establishes session with server and binds both connections
- ***(IF) 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.
- ***(G) ip tcp intercept watch-timeout <sec>***
  If peers do not negotiate within this time (30 sec) RST is sent
- ***(G) 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
- ***(G) ip tcp intercept drop-mode {oldest | random}***
  By default, the software drops the oldest partial connection.

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
- ***(G) ip reflexive-list timeout <sec>*** - default is 300 sec

Packets initiated by a router are not matched by outbound ACL or any inspection !!!



L3 Security

ACL

- 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.
- ***(G) ip access-list logging interval <msec>*** - 0 means no rate limiting
  (G) ip access-list log-update threshold <count>
- ***(G) ip icmp rate-limit unreachable ...***
  Rate limiting dropped packets when ICMP is generated (administrively prohibited)
- ***(ACL) permit tcp any any {match-all | match-any} +ack +syn -urg -psh ...***
  Match specific bits in TCP packet
- ***(ACL) ... {log | log-input}***
  If ***log-input*** is used, input interface and L2 header information will also be logged
- ***(G) ip access-list resequence <acl> <start> <step>***
  Resequence ACL. By default each entry is seqenced by 10, starting with 10
- Switch ACL
  - PACL control traffic entering a Layer 2 interface. The switch does not support port ACLs in the outbound direction. Supported only on physical interfaces. On a trunk filters traffic on all VLANs present on the trunk port
  - RACL controls routed traffic between VLANs and are applied to L3 interfaces (inbound or outbound)
  - VLAN ACLs (VLAN maps) control all packets (bridged and routed). Packets can either enter the VLAN through a switch port or through a routed port
- IPv4 (0x800) ACL does not catch ARP (0x806), use MAC ACL to filter ARP
- IPv6 uses ICMP Neighbor Discovery, which is implicitly permited in each IPv6 ACL
- Time-based
  - ***time-range <name>***
    absolute start ...
    periodic weekdays ...
  - ***(ACL) permit ip any any time-range <name>***
  - ***showw time-tange*** (check if it's active)

IP Options Drop

- IP Options Selective Drop filter packets with IP options on a router or downstream routers by dropping these packets or ignoring options (watch for RSVP)
- ***(G) ip options {drop | ignore}***
- ***show ip traffic***

uRPF

- The packet must arrive on interface that has the best return path (route) to the source (reverse lookup in the CEF table)
- Unicast RPF is an input function and is applied only on the input interface
- Unicast RPF will allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) functions work properly
- ***(IF) ip verify unicast reverse-path <acl>*** - Legacy way
- ***(IF) 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
- 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 (deny) or forwarded (permit)
- ***show ip interface <if>***

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.
- ***(G) ip source-track <ip-address>***
  Destination address being attacked (configured on a router closest to tracked source)
- ***(G) ip source-track address-limit <number>***
- ***(G) ip source-track syslog-interval <1-1440 min>***
- ***show ip source-track [ip-address] [summary | cache]***

CoPP

- MQC supports named and numbered ACLs, standard and extended
- ***(PM) police rate [burst-normal] [burst-max] conform-action <action> exceed-action <action> [violate-action <action>]***
- ***control-plane [{host | transit | cef-exception}]***
  service-policy {input | output} <name>
  host – trafic directly for router intf (SSH, BGP, EIGRP, SNMP)
  transit – traffic that is software switched by CPU
  cef-exception – CEF switched packets (ARP, BGP, OSPF)
- Port Filter
  - Early dropping (the only action) of packets that are directed toward closed on nonlistened ports on the router
  - ***(G) policy-map type port-filter <name>***
  - ***(G) class-map type port-filter [match-all | match-any] <name>***
  - ***(CM) match {closed-ports | [not] port} {tcp | udp}***
  - ***(IF) service-policy type port-filter {input} <name>***
  - ***show control plane host open ports***
- Queue Threshold
  - ***(G) class-map type queue-threshold [match-all | match-any] <name>***
  - ***match protocol [bgp | dns | ftp | http | igmp | snmp | ssh | syslog | telnet | tftp] [cr]***
  - ***(CM) queue-limit <#>*** - 0-255 packets
- Management Protection
  - Restrict interfaces on which network management packets are allowed
  - Not supported for strict, dedicated OOB management interfaces
  - ***control-plane host***
    management-interface <if> allow <protocols>
  - ***show management-interface***

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

L2 Security

DHCP snooping

Prevents server spoofing and pool exchaution attack

(G) ip dhcp snooping vlan <#> [smartlog]
Enable snooping on specific VLAN. Smartlog sends content of dropped packets to NetFlow collector

(G) ip dhcp snooping

(G) ip dhcp snooping database write-delay <sec>
Specify the duration for which the transfer should be delayed (default 300) after the binding database changes

(IF) ip dhcp snooping trust

(G) ip dhcp snooping information option allow-untrusted
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.

(G) ip dhcp snooping database <filesystem>
By default all entries are removed if switch is reloaded. Dynamic and static entries can be stored in external DB.

(G) ip dhcp snooping database timeout <sec>
Specify (default 300) how long to wait for the database transfer process to finish before stopping the process

(IF) ip dhcp snooping limit rate <#>
No limit by default. No more than 100 is recommended on untrusted interfaces

(G) no ip dhcp relay information option
Disable (enabled by default) inserting and removing Option-82 field (by the switch). Option-82 adds circuit-id (port ID) and remote-id (switch ID). Must be set on each switch. 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 verify mac-address
Verify that the source MAC in a DHCP packet received on untrusted ports matches the client hardware address in the packet. The default is to verify that the source MAC address matches the client hardware address in the packet.

(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.

show ip dhcp snooping [database | binding | statistics]

show ip source binding

DAI

Dynamic ARP Inspection – prevents ARP poisoning attacks

ARP ACLs take precedence over entries in the DHCP snooping binding database. The switch first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the switch also denies the packet even if a valid binding exists in the database populated by DHCP snooping

Dynamic ARP inspection is an ingress security feature; it does not perform any egress checking

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

Static

- ***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.
- ***(G) ip arp inspection filter <ARP-acl> vlan <range> [static]***
  DHCP snooping is not required/used if ***static*** keyword is used. Otherwise, ACL is checked first, then DHCP

(G) ip arp inspection validate [src-mac] [dst-mac] [ip]

(G) ip arp inspection limit {rate <pps> [burst <intv>] | none}
Default 15pps/1sec

(G) ip arp inspection log-buffer {entries <#> | logs <#> interval <sec>}
Default 32 entries, 5 messages every 1 sec

(G) ip arp inspection vlan <range> logging {acl-match {matchlog | none} | dhcp-bindings {all | none | permit}}
Control the type of packets that are logged per VLAN. By default, all denied or all dropped packets are logged

show ip arp inspection interfaces

show ip arp inspection vlan

show arp access-list

802.1x

Until the device is authenticated, 802.1x allows only Extensible Authentication Protocol over LAN (EAPOL)

Supplicant – client device that requests network access

Authenticator – network device (switch) that serves Supplicant's authorization requests

Authentication Server – server (RADIUS) providing authentication services

(G) dot1x system auth-control
Enable dot1x (required)

(G) aaa authentication dot1x group ...

(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.

(IF) dot1x guest-vlan <vlan-id>
The switch assigns clients to a guest VLAN when it does not receive a response to EAPOL

Multi-Domain Auth (MDA) allows IP Phone and a PC to authenticate on the same port (separate Voice and Data VLANs)

(IF) dot1x host-mode {single-host | multi-host | multi-domain}
multi-host – allow multiple hosts after a single host has been authenticated
multi-domain – allow host and voice device to be authenticated

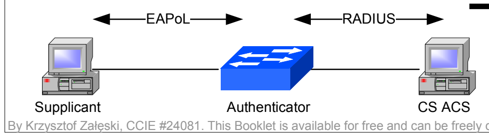
(IF) dot1x auth-fail vlan <vlan-id>
Define restricted vlan upon authentication failure. The user is not notified of the authentication failure.

MDA can use MAC authentication bypass as a fallback mechanism to allow the switch port to connect to devices that do not support IEEE 802.1x authentication

(G) dot1x reauthentication [interface <intf>]
Re-enable authentication on restricted vlan (exec mode)

(G) dot1x timeout reauth-period <sec>
Re-authentication period for restricted vlan

show dot1x interface <if> details



IP Source Guard

Not supported on EtherChannels

DHCP snooping extension used to prevent attacks when a host tries to use other host's IP

When enabled, the switch initially blocks all IP traffic on an interface except for DHCP packets. PACL is applied to the interface, which allows only IP traffic with a source IP address in the IP source binding table. That ACL takes precedence over any RACLs or VLAN maps that affect the same interface

DHCP snooping must be enabled on the access VLAN to which the interface belongs

(IF) ip verify source [smartlog]

(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. The DHCP server must support option 82, or the client is not assigned an IP address. The MAC address in the DHCP packet is not learned as a secure address. The MAC address of the DHCP client is learned as a secure address only when the switch receives non-DHCP data traffic

(G) 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

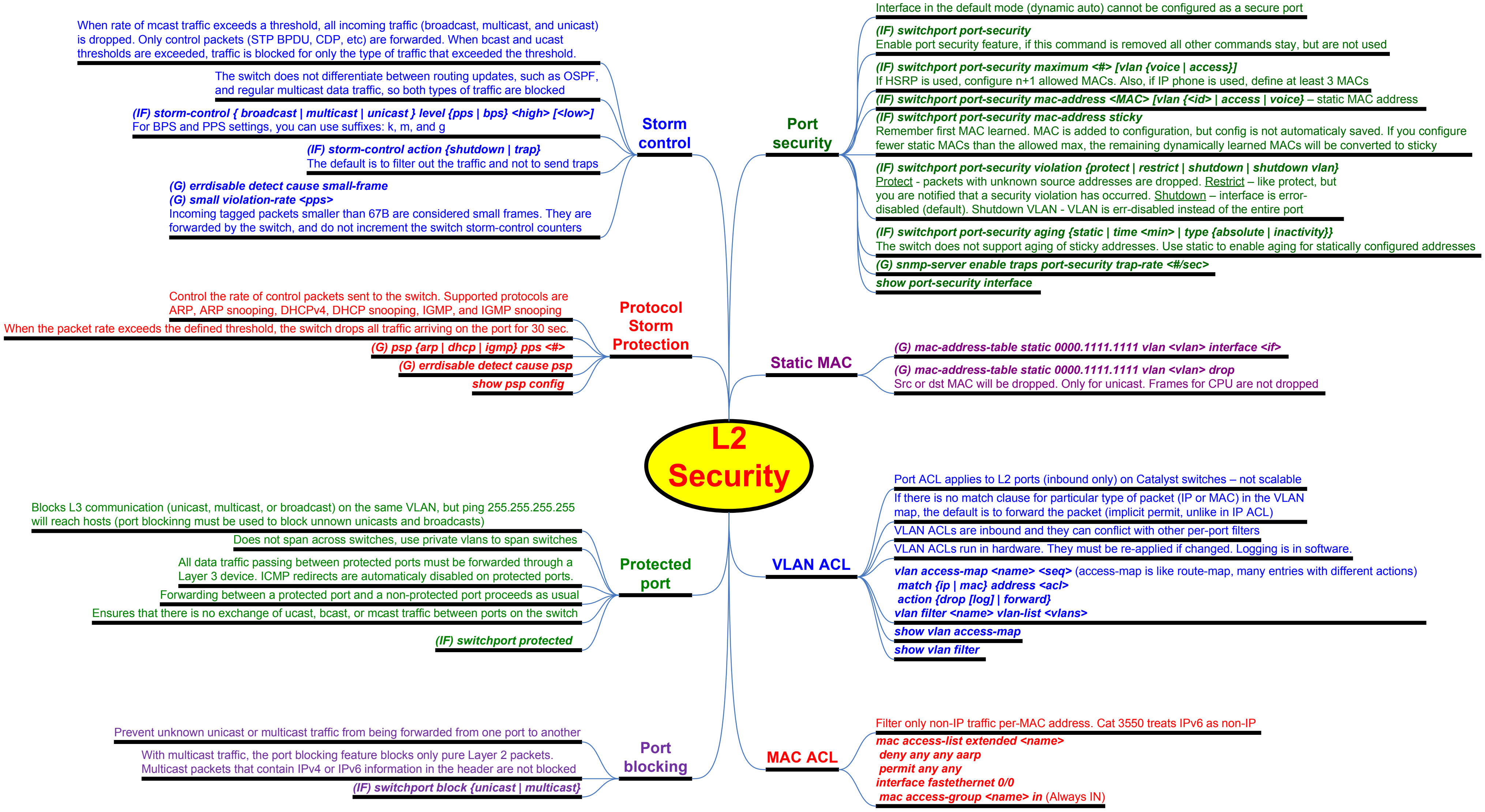
Static hosts

- ***(G) ip device tracking***
  Turn on the IP host table, and globally enable IP device tracking
- ***(IF) ip verify source tracking port-security***
  Enable IPSG for static hosts with MAC address filtering
- ***(IF) ip device tracking maximum <#>***
  Set the number of static IPs allowed on the port. Like Port-Security in L3

show ip verify source

show ip source binding

show ip device track all



L2 Security

Storm control

When rate of mcast traffic exceeds a threshold, all incoming traffic (broadcast, multicast, and unicast) is dropped. Only control packets (STP BPDU, CDP, etc) are forwarded. When bcast and ucast thresholds are exceeded, traffic is blocked for only the type of traffic that exceeded the threshold.

The switch does not differentiate between routing updates, such as OSPF, and regular multicast data traffic, so both types of traffic are blocked

(IF) storm-control { broadcast | multicast | unicast } level {pps | bps} <high> [<low>]
For BPS and PPS settings, you can use suffixes: k, m, and g

(IF) storm-control action {shutdown | trap}
The default is to filter out the traffic and not to send traps

(G) errdisable detect cause small-frame
(G) small violation-rate <pps>
Incoming tagged packets smaller than 67B are considered small frames. They are forwarded by the switch, and do not increment the switch storm-control counters

Protocol Storm Protection

Control the rate of control packets sent to the switch. Supported protocols are ARP, ARP snooping, DHCPv4, DHCP snooping, IGMP, and IGMP snooping

When the packet rate exceeds the defined threshold, the switch drops all traffic arriving on the port for 30 sec.

(G) psp {arp | dhcp | igmp} pps <#>

(G) errdisable detect cause psp

show psp config

Protected port

Blocks L3 communication (unicast, multicast, or broadcast) on the same VLAN, but ping 255.255.255.255 will reach hosts (port blockinng must be used to block unnown unicasts and broadcasts)

Does not span across switches, use private vlans to span switches

All data traffic passing between protected port 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

Ensures that there is no exchange of ucast, bcast, or mcast traffic between ports on the switch

(IF) switchport protected

Port blocking

Prevent unknown unicast or multicast traffic from being forwarded from one port to another

With multicast traffic, the port blocking feature blocks only pure Layer 2 packets. Multicast packets that contain IPv4 or IPv6 information in the header are not blocked

(IF) switchport block {unicast | multicast}

Port security

Interface in the default mode (dynamic auto) cannot be configured as a secure port

(IF) switchport port-security
Enable port security feature, if this command is removed all other commands stay, but are not used

(IF) 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

(IF) switchport port-security mac-address <MAC> [vlan {<id> | access | voice} – static MAC address

(IF) switchport port-security mac-address sticky
Remember first MAC learned. MAC is added to configuration, but config is not automaticaly saved. If you configure fewer static MACs than the allowed max, the remaining dynamically learned MACs will be converted to sticky

(IF) switchport port-security violation {protect | restrict | shutdown | shutdown vlan}
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). Shutdown VLAN - VLAN is err-disabled instead of the entire port

(IF) switchport port-security aging {static | time <min> | type {absolute | inactivity}}
The switch does not support aging of sticky addresses. Use static to enable aging for statically configured addresses

(G) snmp-server enable traps port-security trap-rate <#/sec>

show port-security interface

Static MAC

(G) mac-address-table static 0000.1111.1111 vlan <vlan> interface <if>

(G) mac-address-table static 0000.1111.1111 vlan <vlan> drop
Src or dst MAC will be dropped. Only for unicast. Frames for CPU are not dropped

VLAN ACL

Port ACL applies to L2 ports (inbound only) on Catalyst switches – not scalable

If there is no match clause for particular type of packet (IP or MAC) in the VLAN map, the default is to forward the packet (implicit permit, unlike in IP 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. Logging is in software.

vlan access-map <name> <seq> (access-map is like route-map, many entries with different actions)
 match {ip | mac} address <acl>
 action {drop [log] | forward}
vlan filter <name> vlan-list <vlans>

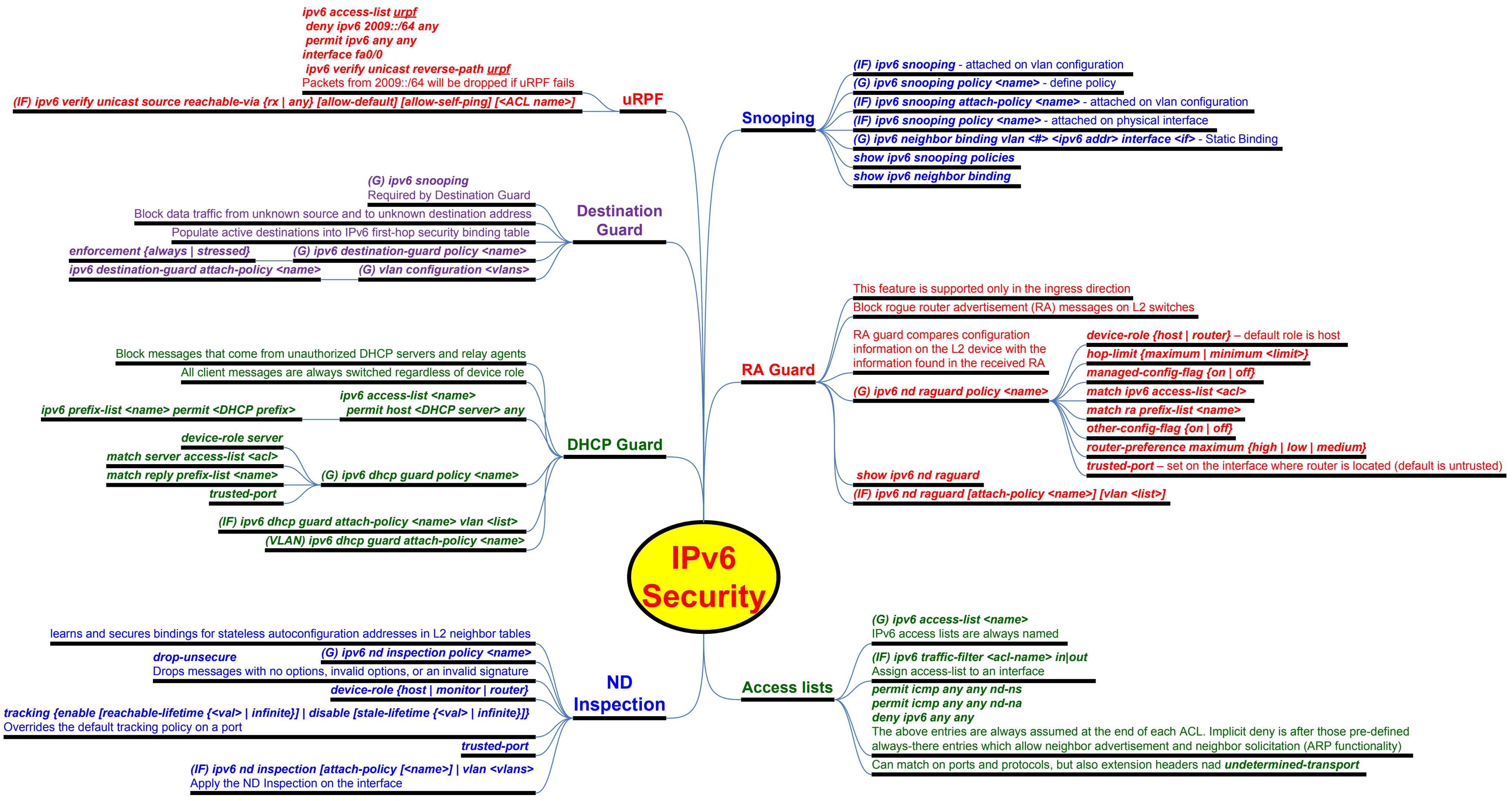
show vlan access-map

show vlan filter

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)



IPv6 Security

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>]

Destination Guard

(G) ipv6 snooping
Required by Destination Guard

Block data traffic from unknown source and to unknown destination address

Populate active destinations into IPv6 first-hop security binding table

(G) ipv6 destination-guard policy <name>
- *enforcement {always | stressed}*

(G) vlan configuration <vlans>
- *ipv6 destination-guard attach-policy <name>*

DHCP Guard

Block messages that come from unauthorized DHCP servers and relay agents

All client messages are always switched regardless of device role

ipv6 access-list <name>
permit host <DHCP server> any
- *ipv6 prefix-list <name> permit <DHCP prefix>*

(G) ipv6 dhcp guard policy <name>
- *device-role server*
- *match server access-list <acl>*
- *match reply prefix-list <name>*
- *trusted-port*

(IF) ipv6 dhcp guard attach-policy <name> vlan <list>

(VLAN) ipv6 dhcp guard attach-policy <name>

ND Inspection

learns and secures bindings for stateless autoconfiguration addresses in L2 neighbor tables

(G) ipv6 nd inspection policy <name>

drop-unsecure
Drops messages with no options, invalid options, or an invalid signature

device-role {host | monitor | router}

tracking {enable [reachable-lifetime {<val> | infinite}] | disable [stale-lifetime {<val> | infinite}]}
Overrides the default tracking policy on a port

trusted-port

(IF) ipv6 nd inspection [attach-policy [<name>] | vlan <vlans>
Apply the ND Inspection on the interface

Snooping

(IF) ipv6 snooping - attached on vlan configuration

(G) ipv6 snooping policy <name> - define policy

(IF) ipv6 snooping attach-policy <name> - attached on vlan configuration

(IF) ipv6 snooping policy <name> - attached on physical interface

(G) ipv6 neighbor binding vlan <#> <ipv6 addr> interface <if> - Static Binding

show ipv6 snooping policies

show ipv6 neighbor binding

RA Guard

This feature is supported only in the ingress direction

Block rogue router advertisement (RA) messages on L2 switches

RA guard compares configuration information on the L2 device with the information found in the received RA

(G) ipv6 nd raguard policy <name>
- *device-role {host | router}* – default role is host
- *hop-limit {maximum | minimum <limit>}*
- *managed-config-flag {on | off}*
- *match ipv6 access-list <acl>*
- *match ra prefix-list <name>*
- *other-config-flag {on | off}*
- *router-preference maximum {high | low | medium}*
- *trusted-port* – set on the interface where router is located (default is untrusted)

show ipv6 nd raguard

(IF) ipv6 nd raguard [attach-policy <name>] [vlan <list>]

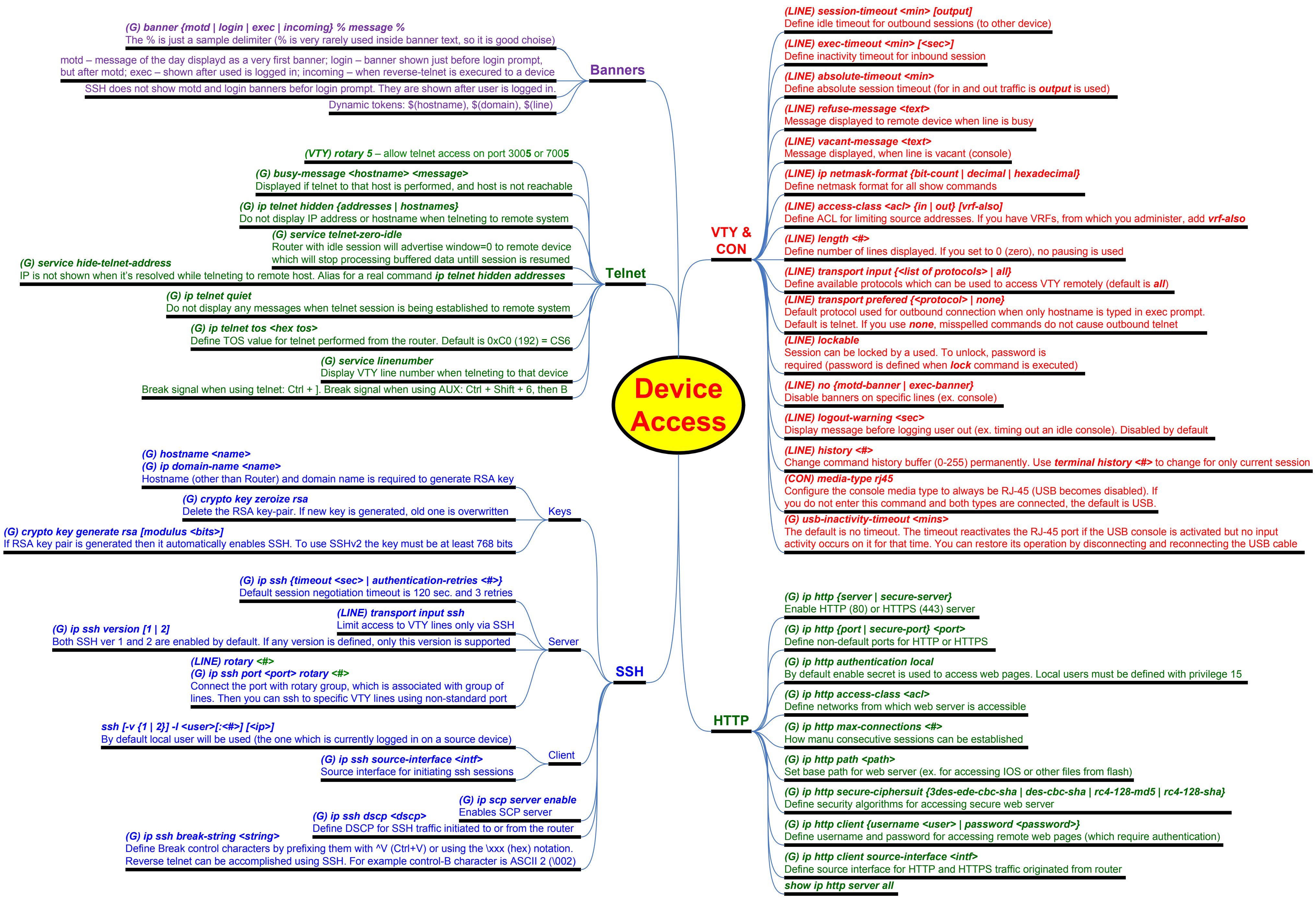
Access lists

(G) ipv6 access-list <name>
IPv6 access lists are always named

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

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

Can match on ports and protocols, but also extension headers nad ***undetermined-transport***



Device Access

Banners

(G) banner {motd | login | exec | incoming} % message %
The % is just a sample delimiter (% is very rarely used inside banner text, so it is good choise)

motd – message of the day displayd as a very first banner; login – banner shown just before login prompt, but after motd; exec – shown after used is logged in; incoming – when reverse-telnet is execured to a device

SSH does not show motd and login banners befor login prompt. They are shown after user is logged in.

Dynamic tokens: $(hostname), $(domain), $(line)

Telnet

(VTY) rotary 5 – allow telnet access on port 300**5** or 700**5**

(G) busy-message <hostname> <message>
Displayed if telnet to that host is performed, and host is not reachable

(G) ip telnet hidden {addresses | hostnames}
Do not display IP address or hostname when telneting to remote system

(G) service telnet-zero-idle
Router with idle session will advertise window=0 to remote device which will stop processing buffered data untill session is resumed

(G) service hide-telnet-address
IP is not shown when it's resolved while telneting to remote host. Alias for a real command ***ip telnet hidden addresses***

(G) ip telnet quiet
Do not display any messages when telnet session is being established to remote system

(G) ip telnet tos <hex tos>
Define TOS value for telnet performed from the router. Default is 0xC0 (192) = CS6

(G) service linenumber
Display VTY line number when telneting to that device

Break signal when using telnet: Ctrl +]. Break signal when using AUX: Ctrl + Shift + 6, then B

SSH

Keys

(G) hostname <name>
(G) ip domain-name <name>
Hostname (other than Router) and domain name is required to generate RSA key

(G) crypto key zeroize rsa
Delete the RSA key-pair. If new key is generated, old one is overwritten

(G) crypto key generate rsa [modulus <bits>]
If RSA key pair is generated then it automatically enables SSH. To use SSHv2 the key must be at least 768 bits

Server

(G) ip ssh {timeout <sec> | authentication-retries <#>}
Default session negotiation timeout is 120 sec. and 3 retries

(LINE) transport input ssh
Limit access to VTY lines only via SSH

(G) 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

(LINE) rotary <#>
(G) ip ssh port <port> rotary <#>
Connect the port with rotary group, which is associated with group of lines. Then you can ssh to specific VTY lines using non-standard port

Client

ssh [-v {1 | 2}] -l <user>[:<#>] [<ip>]
By default local user will be used (the one which is currently logged in on a source device)

(G) ip ssh source-interface <intf>
Source interface for initiating ssh sessions

(G) ip scp server enable
Enables SCP server

(G) ip ssh dscp <dscp>
Define DSCP for SSH traffic initiated to or from the router

(G) ip ssh break-string <string>
Define Break control characters by prefixing them with ^V (Ctrl+V) or using the \xxx (hex) notation. Reverse telnet can be accomplished using SSH. For example control-B character is ASCII 2 (\002)

VTY & CON

(LINE) session-timeout <min> [output]
Define idle timeout for outbound sessions (to other device)

(LINE) exec-timeout <min> [<sec>]
Define inactivity timeout for inbound session

(LINE) absolute-timeout <min>
Define absolute session timeout (for in and out traffic is ***output*** is used)

(LINE) refuse-message <text>
Message displayed to remote device when line is busy

(LINE) vacant-message <text>
Message displayed, when line is vacant (console)

(LINE) ip netmask-format {bit-count | decimal | hexadecimal}
Define netmask format for all show commands

(LINE) access-class <acl> {in | out} [vrf-also]
Define ACL for limiting source addresses. If you have VRFs, from which you administer, add ***vrf-also***

(LINE) length <#>
Define number of lines displayed. If you set to 0 (zero), no pausing is used

(LINE) transport input {<list of protocols> | all}
Define available protocols which can be used to access VTY remotely (default is ***all***)

(LINE) transport prefered {<protocol> | none}
Default protocol used for outbound connection when only hostname is typed in exec prompt. Default is telnet. If you use ***none***, misspelled commands do not cause outbound telnet

(LINE) lockable
Session can be locked by a used. To unlock, password is required (password is defined when ***lock*** command is executed)

(LINE) no {motd-banner | exec-banner}
Disable banners on specific lines (ex. console)

(LINE) logout-warning <sec>
Display message before logging user out (ex. timing out an idle console). Disabled by default

(LINE) history <#>
Change command history buffer (0-255) permanently. Use ***terminal history <#>*** to change for only current session

(CON) media-type rj45
Configure the console media type to always be RJ-45 (USB becomes disabled). If you do not enter this command and both types are connected, the default is USB.

(G) usb-inactivity-timeout <mins>
The default is no timeout. The timeout reactivates the RJ-45 port if the USB console is activated but no input activity occurs on it for that time. You can restore its operation by disconnecting and reconnecting the USB cable

HTTP

(G) ip http {server | secure-server}
Enable HTTP (80) or HTTPS (443) server

(G) ip http {port | secure-port} <port>
Define non-default ports for HTTP or HTTPS

(G) ip http authentication local
By default enable secret is used to access web pages. Local users must be defined with privilege 15

(G) ip http access-class <acl>
Define networks from which web server is accessible

(G) ip http max-connections <#>
How manu consecutive sessions can be established

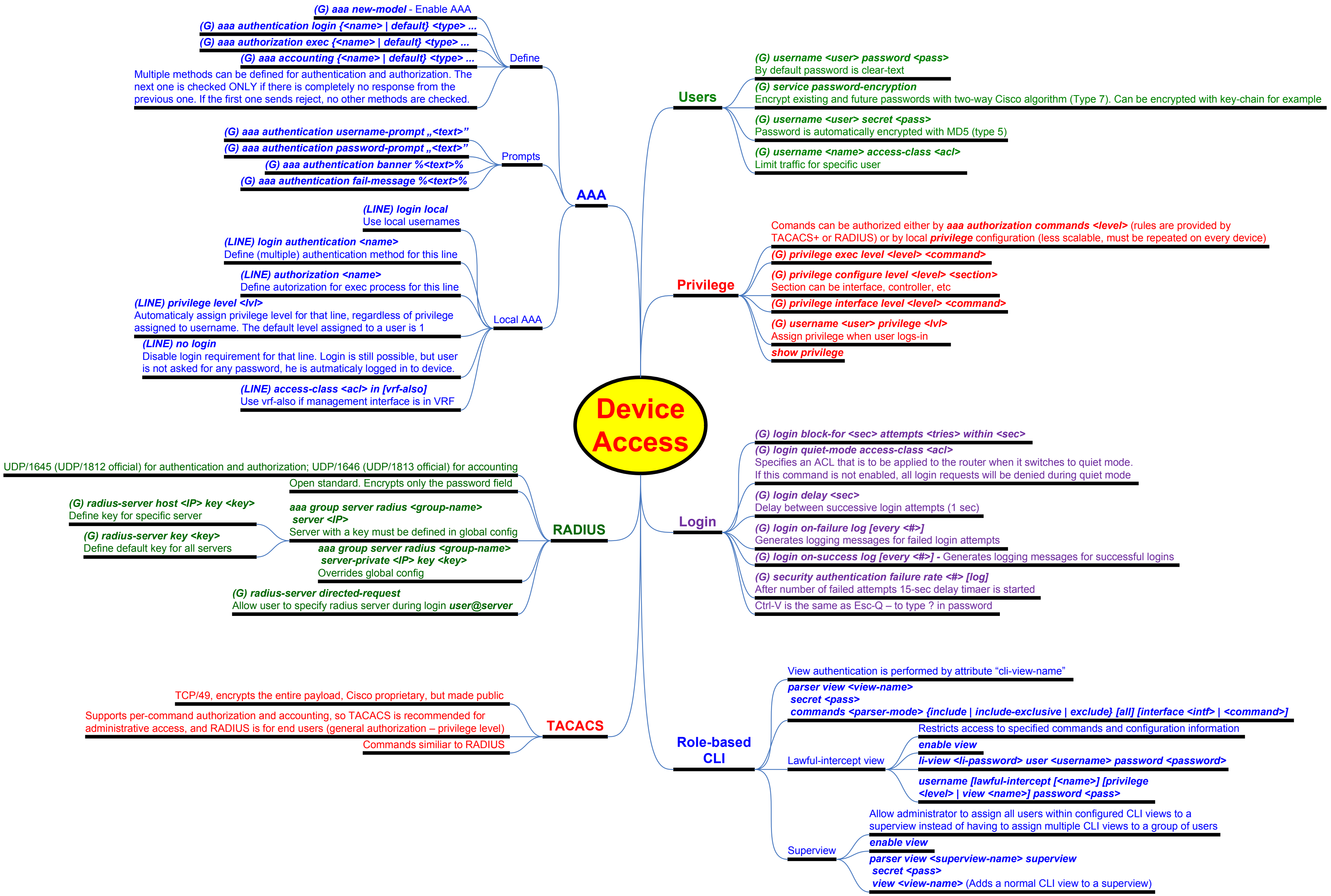
(G) ip http path <path>
Set base path for web server (ex. for accessing IOS or other files from flash)

(G) ip http secure-ciphersuit {3des-ede-cbc-sha | des-cbc-sha | rc4-128-md5 | rc4-128-sha}
Define security algorithms for accessing secure web server

(G) ip http client {username <user> | password <password>}
Define username and password for accessing remote web pages (which require authentication)

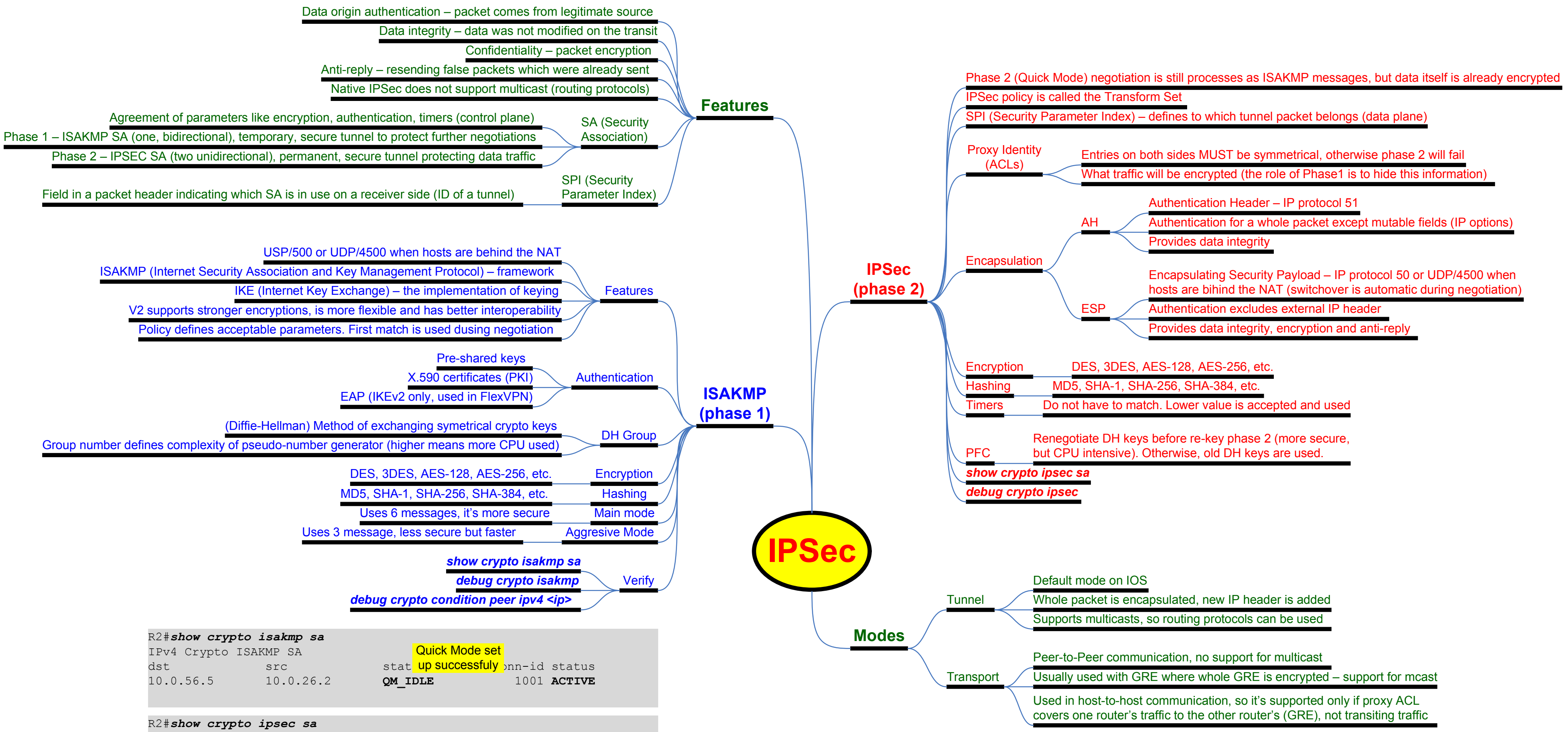
(G) ip http client source-interface <intf>
Define source interface for HTTP and HTTPS traffic originated from router

show ip http server all



Device Access

- **AAA**
  - Define
    - ***(G) aaa new-model*** - Enable AAA
    - ***(G) aaa authentication login {<name> | default} <type> ...***
    - ***(G) aaa authorization exec {<name> | default} <type> ...***
    - ***(G) 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
    - ***(G) aaa authentication username-prompt „<text>"***
    - ***(G) aaa authentication password-prompt „<text>"***
    - ***(G) aaa authentication banner %<text>%***
    - ***(G) aaa authentication fail-message %<text>%***
  - Local AAA
    - ***(LINE) login local***
Use local usernames
    - ***(LINE) login authentication <name>***
Define (multiple) authentication method for this line
    - ***(LINE) authorization <name>***
Define autorization for exec process for this line
    - ***(LINE) privilege level <lvl>***
Automaticaly assign privilege level for that line, regardless of privilege assigned to username. The default level assigned to a user is 1
    - ***(LINE) no login***
Disable login requirement for that line. Login is still possible, but user is not asked for any password, he is autmaticaly logged in to device.
    - ***(LINE) access-class <acl> in [vrf-also]***
Use vrf-also if management interface is in VRF
- **RADIUS**
  - UDP/1645 (UDP/1812 official) for authentication and authorization; UDP/1646 (UDP/1813 official) for accounting
  - Open standard. Encrypts only the password field
  - ***aaa group server radius <group-name>***
server <IP>
Server with a key must be defined in global config
    - ***(G) radius-server host <IP> key <key>***
Define key for specific server
    - ***(G) radius-server key <key>***
Define default key for all servers
  - ***aaa group server radius <group-name>***
server-private <IP> key <key>
Overrides global config
  - ***(G) radius-server directed-request***
Allow user to specify radius server during login ***user@server***
- **TACACS**
  - TCP/49, encrypts the entire payload, Cisco proprietary, but made public
  - Supports per-command authorization and accounting, so TACACS is recommended for administrative access, and RADIUS is for end users (general authorization – privilege level)
  - Commands similiar to RADIUS
- **Users**
  - ***(G) username <user> password <pass>***
By default password is clear-text
  - ***(G) service password-encryption***
Encrypt existing and future passwords with two-way Cisco algorithm (Type 7). Can be encrypted with key-chain for example
  - ***(G) username <user> secret <pass>***
Password is automatically encrypted with MD5 (type 5)
  - ***(G) username <name> access-class <acl>***
Limit traffic for specific user
- **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)
  - ***(G) privilege exec level <level> <command>***
  - ***(G) privilege configure level <level> <section>***
Section can be interface, controller, etc
  - ***(G) privilege interface level <level> <command>***
  - ***(G) username <user> privilege <lvl>***
Assign privilege when user logs-in
  - ***show privilege***
- **Login**
  - ***(G) login block-for <sec> attempts <tries> within <sec>***
  - ***(G) 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
  - ***(G) login delay <sec>***
Delay between successive login attempts (1 sec)
  - ***(G) login on-failure log [every <#>]***
Generates logging messages for failed login attempts
  - ***(G) login on-success log [every <#>] -*** Generates logging messages for successful logins
  - ***(G) 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
- **Role-based CLI**
  - View authentication is performed by attribute "cli-view-name"
  - ***parser view <view-name>***
secret <pass>
commands <parser-mode> {include | include-exclusive | exclude} [all] [interface <intf> | <command>]
  - 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)



IPSec

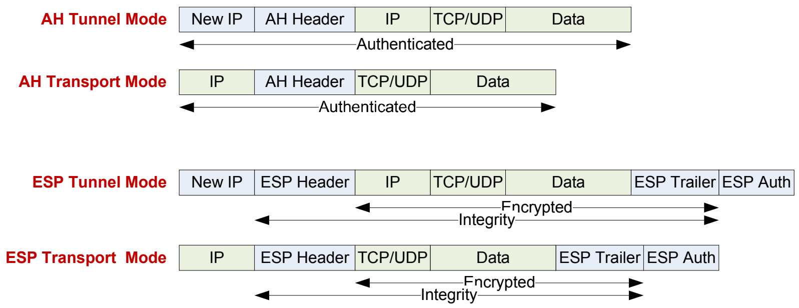
- **Features**
  - Data origin authentication – packet comes from legitimate source
  - Data integrity – data was not modified on the transit
  - Confidentiality – packet encryption
  - Anti-reply – resending false packets which were already sent
  - Native IPSec does not support multicast (routing protocols)
  - SA (Security Association)
    - Agreement of parameters like encryption, authentication, timers (control plane)
    - Phase 1 – ISAKMP SA (one, bidirectional), temporary, secure tunnel to protect further negotiations
    - Phase 2 – IPSEC SA (two unidirectional), permanent, secure tunnel protecting data traffic
  - SPI (Security Parameter Index)
    - Field in a packet header indicating which SA is in use on a receiver side (ID of a tunnel)
- **ISAKMP (phase 1)**
  - Features
    - USP/500 or UDP/4500 when hosts are behind the NAT
    - ISAKMP (Internet Security Association and Key Management Protocol) – framework
    - IKE (Internet Key Exchange) – the implementation of keying
    - V2 supports stronger encryptions, is more flexible and has better interoperability
    - Policy defines acceptable parameters. First match is used dusing negotiation
  - Authentication
    - Pre-shared keys
    - X.590 certificates (PKI)
    - EAP (IKEv2 only, used in FlexVPN)
  - DH Group
    - (Diffie-Hellman) Method of exchanging symetrical crypto keys
    - Group number defines complexity of pseudo-number generator (higher means more CPU used)
  - Encryption – DES, 3DES, AES-128, AES-256, etc.
  - Hashing – MD5, SHA-1, SHA-256, SHA-384, etc.
  - Main mode – Uses 6 messages, it's more secure
  - Aggresive Mode – Uses 3 message, less secure but faster
  - Verify
    - *show crypto isakmp sa*
    - *debug crypto isakmp*
    - *debug crypto condition peer ipv4 <ip>*
- **IPSec (phase 2)**
  - Phase 2 (Quick Mode) negotiation is still processes as ISAKMP messages, but data itself is already encrypted
  - IPSec policy is called the Transform Set
  - SPI (Security Parameter Index) – defines to which tunnel packet belongs (data plane)
  - Proxy Identity (ACLs)
    - Entries on both sides MUST be symmetrical, otherwise phase 2 will fail
    - What traffic will be encrypted (the role of Phase1 is to hide this information)
  - Encapsulation
    - AH
      - Authentication Header – IP protocol 51
      - Authentication for a whole packet except mutable fields (IP options)
      - Provides data integrity
    - ESP
      - Encapsulating Security Payload – IP protocol 50 or UDP/4500 when hosts are bihind the NAT (switchover is automatic during negotiation)
      - Authentication excludes external IP header
      - Provides data integrity, encryption and anti-reply
  - Encryption – DES, 3DES, AES-128, AES-256, etc.
  - Hashing – MD5, SHA-1, SHA-256, SHA-384, etc.
  - Timers – Do not have to match. Lower value is accepted and used
  - PFC – Renegotiate DH keys before re-key phase 2 (more secure, but CPU intensive). Otherwise, old DH keys are used.
  - *show crypto ipsec sa*
  - *debug crypto ipsec*
- **Modes**
  - Tunnel
    - Default mode on IOS
    - Whole packet is encapsulated, new IP header is added
    - Supports multicasts, so routing protocols can be used
  - Transport
    - Peer-to-Peer communication, no support for multicast
    - Usually used with GRE where whole GRE is encrypted – support for mcast
    - Used in host-to-host communication, so it's supported only if proxy ACL covers one router's traffic to the other router's (GRE), not transiting traffic

```
R2#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
10.0.56.5       10.0.26.2       QM_IDLE           1001 ACTIVE
```

Quick Mode set up successfuly

```
R2#show crypto ipsec sa

interface: GigabitEthernet1/0
    Crypto map tag: SPOKE, local addr 10.0.26.2
   [...]
   local  ident (addr/mask/prot/port): (2.2.2.2/255.255.255.255/1/0)
   remote ident (addr/mask/prot/port): (5.5.5.5/255.255.255.255/1/0)
   current_peer 10.0.56.5 port 500
   [...]
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest: 10
    #pkts decaps: 9, #pkts decrypt: 9, #pkts verify: 9
   [...]
     local crypto endpt.: 10.0.26.2, remote crypto endpt.: 10.0.56.5
     path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1/0
     current outbound spi: 0xF9500466(4182770790)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
   [...]
        Status: ACTIVE(ACTIVE)
     inbound ah sas:
   [...]
        Status: ACTIVE(
   [...]
```

Outbound SPI should be also Active (bidirectional)



AH Tunnel Mode

New IP	AH Header	IP	TCP/UDP	Data

Authenticated

AH Transport Mode

IP	AH Header	TCP/UDP	Data

Authenticated

ESP Tunnel Mode

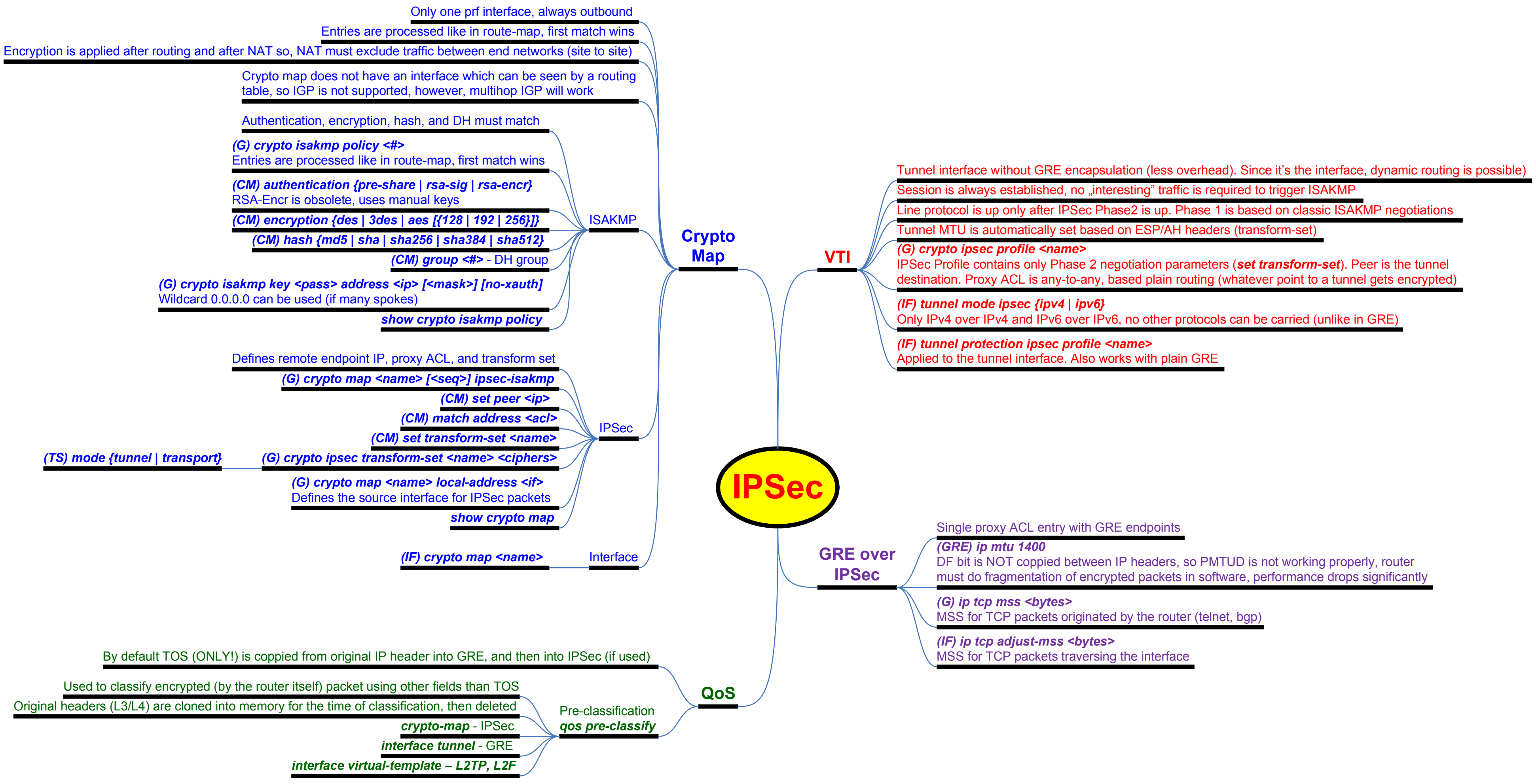
New IP	ESP Header	IP	TCP/UDP	Data	ESP Trailer	ESP Auth

Encrypted
Integrity

ESP Transport Mode

IP	ESP Header	TCP/UDP	Data	ESP Trailer	ESP Auth

Encrypted
Integrity



- **IPSec**
  - **Crypto Map**
    - Only one prf interface, always outbound
    - Entries are processed like in route-map, first match wins
    - Encryption is applied after routing and after NAT so, NAT must exclude traffic between end networks (site to site)
    - Crypto map does not have an interface which can be seen by a routing table, so IGP is not supported, however, multihop IGP will work
    - ISAKMP
      - Authentication, encryption, hash, and DH must match
      - ***(G) crypto isakmp policy <#>***
        Entries are processed like in route-map, first match wins
      - ***(CM) authentication {pre-share | rsa-sig | rsa-encr}***
        RSA-Encr is obsolete, uses manual keys
      - ***(CM) encryption {des | 3des | aes [{128 | 192 | 256}]}***
      - ***(CM) hash {md5 | sha | sha256 | sha384 | sha512}***
      - ***(CM) group <#>*** - DH group
      - ***(G) crypto isakmp key <pass> address <ip> [<mask>] [no-xauth]***
        Wildcard 0.0.0.0 can be used (if many spokes)
      - ***show crypto isakmp policy***
    - IPSec
      - Defines remote endpoint IP, proxy ACL, and transform set
      - ***(G) crypto map <name> [<seq>] ipsec-isakmp***
      - ***(CM) set peer <ip>***
      - ***(CM) match address <acl>***
      - ***(CM) set transform-set <name>***
      - ***(G) crypto ipsec transform-set <name> <ciphers>***
        - ***(TS) mode {tunnel | transport}***
      - ***(G) crypto map <name> local-address <if>***
        Defines the source interface for IPSec packets
      - ***show crypto map***
    - Interface
      - ***(IF) crypto map <name>***
  - **QoS**
    - By default TOS (ONLY!) is coppied from original IP header into GRE, and then into IPSec (if used)
    - Pre-classification
      qos pre-classify
      - Used to classify encrypted (by the router itself) packet using other fields than TOS
      - Original headers (L3/L4) are cloned into memory for the time of classification, then deleted
      - ***crypto-map*** - IPSec
      - ***interface tunnel*** - GRE
      - ***interface virtual-template – L2TP, L2F***
  - **VTI**
    - Tunnel interface without GRE encapsulation (less overhead). Since it's the interface, dynamic routing is possible)
    - Session is always established, no „interesting" traffic is required to trigger ISAKMP
    - Line protocol is up only after IPSec Phase2 is up. Phase 1 is based on classic ISAKMP negotiations
    - Tunnel MTU is automatically set based on ESP/AH headers (transform-set)
    - ***(G) crypto ipsec profile <name>***
      IPSec Profile contains only Phase 2 negotiation parameters (***set transform-set***). Peer is the tunnel destination. Proxy ACL is any-to-any, based plain routing (whatever point to a tunnel gets encrypted)
    - ***(IF) tunnel mode ipsec {ipv4 | ipv6}***
      Only IPv4 over IPv4 and IPv6 over IPv6, no other protocols can be carried (unlike in GRE)
    - ***(IF) tunnel protection ipsec profile <name>***
      Applied to the tunnel interface. Also works with plain GRE
  - **GRE over IPSec**
    - Single proxy ACL entry with GRE endpoints
    - ***(GRE) ip mtu 1400***
      DF bit is NOT coppied between IP headers, so PMTUD is not working properly, router must do fragmentation of encrypted packets in software, performance drops significantly
    - ***(G) ip tcp mss <bytes>***
      MSS for TCP packets originated by the router (telnet, bgp)
    - ***(IF) ip tcp adjust-mss <bytes>***
      MSS for TCP packets traversing the interface

DMVPN

Features

- Dynamic spoke-to-spoke tunnel creation. Independent of service provider, can be run over the Internet
- Large-scale scalable VPN implementation with single mGRE (protocol 47) interface
- NHRP is used to discover endpoints. The hub (NHRP Server) is responsible for mappings
- Underlay (NBMA) protocols are used for endpoint reachability (MPLS, Internet). Overlay protocols exchange customer's networks
- EIGRP and BGP are recommended as overlay protocols. OSPF does not scale that much (flooding)
- Tunnels from spoke to the hub are permanent. Dynamic spoke-to-spoke tunnels are established and torn down based on traffic patterns. They are not permanent.
- Spokes know other spokes internal IPs via overlay routing protocols
- DMVPN does not support multicast, it's a replicated unicast to spokes (underlying network). However, mcast packets are encapsulated inside GRE tunnels
- Mcast spoke-to-spoke is not supported (no control protocol which could signal membership in DMVPN
- Encryption of mGRE is optional

NHRP

- NHRP is send inside GRE tunnel (protocol 0x2001)
- Next Hop Resolution Protocol – spokes can have DHCP/dynamic IP addresses and still register to the hub
- For spoke-to-spoke communication spoke asks the hub for the other spoke's WAN IP
- Registration Request: spoke registes NBMA and WAN addresses to NHS
- Resolution Request: spoke asks NHS for NBMA-to-WAN mapping for the other spoke
- Redirect: NHS redirects traffic going through it to direct spoke-to-spoke traffic. Used only in phase 3
- Spoke-to-spoke tunnels stay up if the hub goes down, but no new tunnels can be created
- ***(mGRE) ip nhrp max-send <pkt-count> every <sec>***
  Max frequency at which NHRP packets can be sent. Default 100 packets in 10 sec

Flags

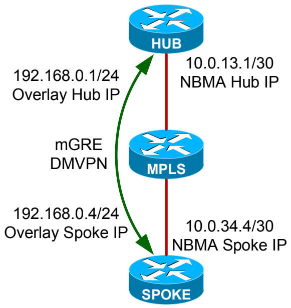
- Authoritative – NHRP information was obtained directly from the NHS
- Implicit – entries learned from an NHRP packet being forwarded or from a request from local router.
- Local – mapping entries that are for networks local to this router
- Nat – NHS client supports NAT extension (spoke is behind a NAT router)
- Negative – initial request (incomplete) suppresses other requests while the resolution is being resolved
- (no socket) – the router is an intermediate node in the path between the two endpoints and we only want to create short-cut tunnels between the initial entrance and final exit point
- Registered – created by an NHRP registration request. Refreshed only by consecutive registrations
- Router – mapping for remote router that is accessing a network behind the remote router
- Unique – NHRP registration requests have the unique flag set
- Used – data packets are process-switched and this mapping entry was used in less than 120 sec)

mGRE

- Hub's mGRE interface is always up
- State of the spoke's interface is determined by successful registration to the hub
- ***(mGRE) tunnel mode gre multipoint***
- ***(mGRE) ip address <ip> <mask>***
  All spokes and the hub must be in common subnem (large LAN)
- ***(mGRE) tunnel key <#>***
  Optional if there are multiple tunnels with separate source addresses. Must be dsed to separate data plane if there are more tunnels using the same source address. Used in GRE header, not NHRP
- ***(mGRE) ip nhrp network-id <#>***
  Optional. Define the NHRP domain if multiple tunnels are on the same router. Local meaning only, not advertised. IDs on different router in the sam cloud do not have to match (like ospf process ID). If tunnel key on two tunnels is not defined, and bot tunnels have the same network-id they are „glued" to form one domain
- ***(mGRE) tunnel source <if>***
  If you do not defined the source interface the line protocol on the tunnel will be down

Hub

- NHRP Server (NHS). Maintains mappings for all spokes
- ***(mGRE) ip nhrp map multicast dynamic***
  The mGRE is a multipoint but not multicast interface. It replicates mcast packets as unicasts. Without this command, routing protocols must use unicast updates (***neighbor*** command)
- ***(mGRE) ip nhrp server-only [non-caching]***
  Do not originate NHRP requests
- ***(IF) ip nhrp holdtime <sec>***
  How long (default 7200 sec.) spokes keep data from authoritative responses. Advertised by the hub. Recommended values are 300-600s



Spoke

- NHRP Client (NHC). Registers with NHS and informs about outside IP (public) to inside IP (NBMA) mapping
- ***(mGRE) ip nhrp nhs <hub overlay IP> [priority <0-255>***
  Specify NHRP server(s). Priority (0 is highest) define the order in which spokes select hubs to establish tunnels
- ***(mGRE) ip nhrp map <hub overlay IP> <hub NMBA IP>***
  Used to defined mapping for the server (hub), but can also be used fo static spoke-to-spoke mapping
- ***(mGRE) ip nhrp map multicast <hub NBMA IP>***
  If spoke needs to send bcast/mcast packet it is replicated as ucast. If more entries are defined then broadcasts packets are replicated to all. If underlying network supports multicast, then use ***destination*** address in the tunnel
- ***(mGRE) ip nhrp registration [timeout <sec> | no-unique]***
  Timeout is between periodic registration messages (max is NHRP holdtime, default 1/3 of holdtime = 40min). The NHS is declared down if no reply is received after 3 retransmissions (7 seconds) – retransmissions sent in 1, 2, 4, 8, 16, 32, 64 sec. Unique mapping means other private-to-the-same-nbma will be rejected. No-unique useseful when IP is assigned periodically via DHCP
- ***(mGRE) ip nhrp interest {<acl> | none}***
  Define which packets trigger NHRP requests. This is only for triggering tunnels, not filtering packets
- ***(mGRE) ip nhrp use <#>***
  How many packets within a minute must be sent to trigger NHTP request (default is 1 = immediate)

Security

- ***(mGRE) ip nhrp authentication <pass>***
  Authentication extension in NHRP header. Type 7 reversible algorithm (like ***enable password***)
- ***(mGRE) tunnel protection ipsec profile <name> shared***
  Encrypt tunnel with IPSec. Shared mode is used if two or more tunnels share the same source interface
- NHRP runs on top of IPSec, so registration will not work untill IPSec is established

VRF Integration

- ***(mGRE) tunnel vrf <name>***
  The tunnel itself is inside local VRF
- ***(mGRE) ip vrf forwarding <name>***
  Data inside the mGRE tunnel runs inside local VRF

DMVPN

Phase 1

mGRE on the hub, and p2p GRE on spokes. NHRP required for spoke registration. Obsolete

Traffic goes to the hub, is decapsulated and decrypted, then hub encrypts and encapsulates the packet to remote spoke. Traceroute goes to hub, then to spoke. No spoke-to-spoke tunnels. Huge performance impact

(mGRE) tunnel mode gre ip
Standard (default) mode for GRE tunnel on spoke side

(mGRE) tunnel destination <hub WAN IP>
Defines Phase 1. Spokes use static destination, no dynamic discovery

Summarization and sending only 0/0 to spokes is supported on hub side, as NH is set to the Hub IP, and spokes do not talk to each other

Split-horizon is not an issue for distance-vector protocols, as spokes do not need other spokes' addresses. If they do, use ***no ip split-horizon*** for RIP or EIGRP

BGP

When using iBGP and route-reflector on the hub (NH is not changed), since tunnels are p2p GRE, the traffic can reach remote spokes through the hub (NH is directly connected on Tunnel)

OSPF

(IF) ip ospf network point-to-multipoint
OSPF treats GRE tunnels as point-to-point where only one neighbor is supported, so the network type has to be changed on hubs and spokes. P2M network modifies NH and sets it to the hub IP

Cost between spokes is the sum of tunnel cost to the hub and from the hub to the other spoke

All hubs and spokes must to be in the same area, so summarization is not supported on the hub. However, you can send 0/0 to spokes, and filter other routes from RIB (not database) on spokes. Not recommended, as spokes still flood „hidden" routes to other routers in the area

Phase 2

```
R4#traceroute 5.5.5.5
Tracing the rout
VRF info: (vrf i                f out name/id)
  1 192.168.0.1 168 msec 152 msec 204 msec
  2 192.168.0.5 196 msec 152 msec 172 msec
R4#traceroute 5.5.5.5
Type escape seque
Tracing the route
VRF info: (vrf ir                 out name/id)
  1 192.168.0.5 148 msec 144 msec 148 msec
```

First packet goes through the hub

Consecutive packets go directly to the spoke

mGRE tunnels on hub and spoke. Obsolete

First packet goes to the hub, hub does the resolution and sends redirect request with NMBA adddress to the spoke, and next packets go between spokes

Summarization and default routing is not allowed on the hub, because NH must be preserved by the hub for spoke-to-spoke communication. If NH points to the hub we do not do resolution for the spoke, and no dynamic tunnels are created (= phase 1)

EIGRP

(mGRE) no ip split-horizon eigrp <as>
Spokes require specific subnets from other spokes to resolve NH

(mGRE) no ip next-hop-self eigrp <as>
DMVPN Phase 2 requires spoke's NH address, not 0.0.0.0 (set by hub)

OSPF

(IF) ip ospf network broadcast
You have to preserve the NH to establish spoke-to-spoke tunnels

(IF) ip ospf priority 0
None of the spokes must be a DR. Only hubs can be DRs, otherwise whole cloud is broken. Setting the highest priority on the hub is not enough, as priority is used only if there is no DR on the network

Phase 3

(mGRE) ip nhrp redirect [timeout <sec>]
Configured on the hub. Shortcut depends on receiving NHRP redirect message. Timeout defines interval the NHRP redirects are sent for the same NBMA source and destination

(mGRE) ip nhrp shortcut
Configured on the spokes. Allows spokes to install the redirects received from the hub

Redirect message is sent by the hub to remote spoke (originating the traffic) to update the routing table and point to the other spoke's IP (NH)

In new version not only FIB (***show ip nhrp***), but also RIB is updated (NHRP becomes a routing protocol)

First packet goes to the hub, hub does the resolution and sends redirect request with NHBA adddress to the spoke, and next packets go between spokes

New entry in routing table appears (H) which is dynamic and times-out when there is no traffic

Spokes in Phase 3 can have only default route and still use direct spoke-to-spoke communication

Summarization on the hub is supported

QoS

QoS on physical interface where spokes' IP addresses are used for matching is not scalable (max 256 classes)

Per-Tunnel QoS is preferred (IOS/IOS-XE on ASR1000) – on hub router only

Egress QoS on the spoke side (also for spoke-spoke communication) must be configured separately (classical HQoS on physical interface)

QoS policy for each spoke (per virtual tunnel) is created dynamically when the spoke registes to the hub (easy config)

Spoke signals to the hub to which policy-group it wants to be assigned (one group per mGRE), but the QoS config is on the hub (NHRP)

Each spoke has own, independent egress shaper on the hub side

1. Create classical CBWFQ policy (child) – ***policy-map 8-class-cbwfq***

2. Create a parent shaper

```
policy-map Shp-2Mb
 class-map class-default
  shape average 2000000
  service-policy 8-class-cbwfq
```

3. Assign a group to the mGRE tunnel. Multiple groups can be assigned

(Tu0) ip nhrp map group Spoke-2Mb service-policy output Shp-2Mb
(Tu0) ip nhrp map group Spoke-4Mb service-policy output Shp-4Mb

4. Spoke can now request the QoS group

(Tu0) ip nhrp group Spoke-2Mb

show ip nhrp group-map

show dmvpn detail

show policy-map multipoint

IPv6

IPv4 transport, but IPv6 inside the mGRE tunnel, since GRE is multiprotocol

(mGRE) tunnel mode gre multipoint ipv6
Native IPv6 transport. Can transport both IPv4 and IPv6. Requires IKEv2 if IPSec is used