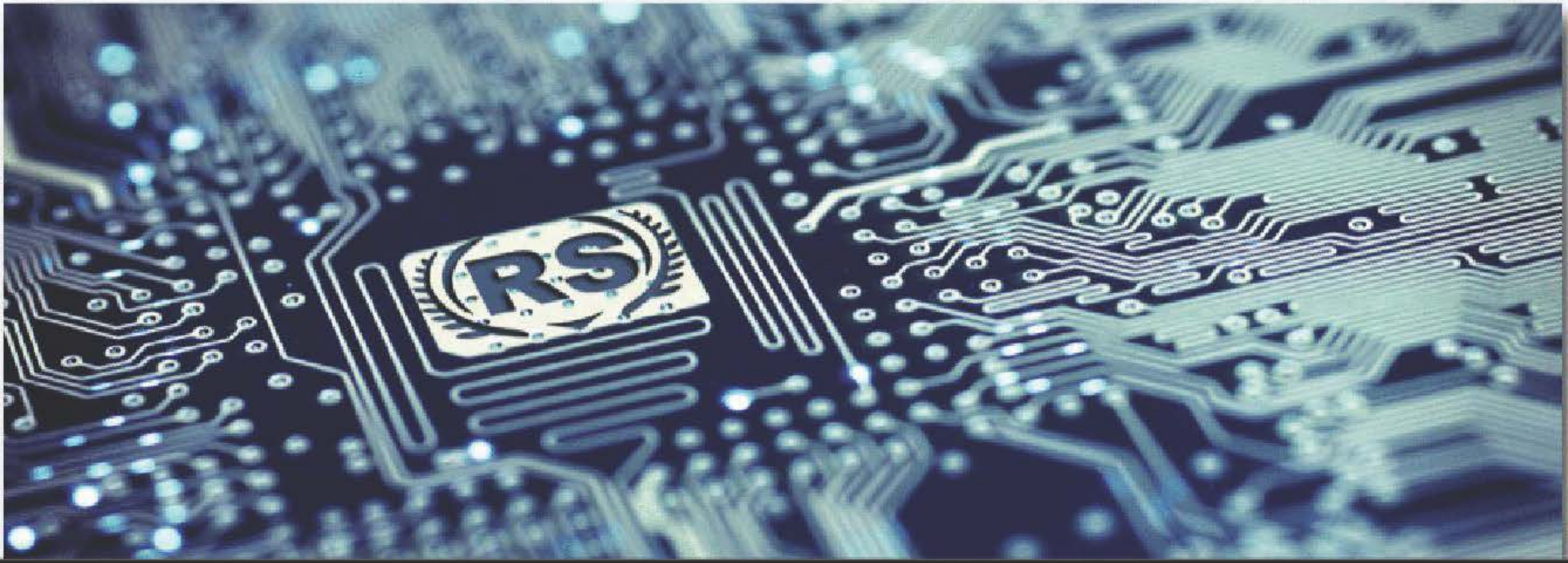
Ruhann du Plessis, CCIE® #24163



CCIE-RS

Routing-Bits Handbook

Routing & Switching

The definitive technology guide for enterprise engineers

routing-bits.com

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

COPYRIGHT INFORMATION

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Routing-Bits Handbook for Routing & Switching
by Ruhann Du Plessis
CCIE #24163 (R&S, SP), CCNP, CCIP
http://www.routing-bits.com

Version 4.3

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

DISCLAIMER

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

This publication, Routing-Bits Handbook for Routing & Switching, is designed to provide technical information and assist candidates in the preparation for CISCO Systems' CCIE Routing and Switching Lab Exam. The information may also assist any networking engineer in his or her day-to-day duties.

While every effort has been made to ensure this book is complete and as accurate as possible, the enclosed information is provided on an 'as is' basis. The author, Routing-Bits, and Cisco Systems, Inc., shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.
This book is NOT sponsored by, endorsed by or affiliated with Cisco Systems, Inc.
Any similarities between the content presented in this book and the actual CCIE lab material is completely coincidental.

```
 _            _
| |          | |
| | ____   __| | _____  _   _
| ||  _ \ / _  ||  __ |( \ / )
| || | | ( (_| || |___| ) v (
|_||_| |_|\____||____) (_/ \_)
```

CHAPTER PAGE

01 – Switching 5

02 – Frame-Relay 33

03 – PPP 45

04 – IP Routing 69

05 – RIP 93

06 – EIGRP 101

07 – OSPF 115

08 – BGP 143

09 – MPLS 177

10 – Multicast 207

11 – IPv6 235

12 – QOS 257

13 – Security 287

14 – Services 313

MOTIVATION FOR THIS BOOK

The main reason that I wrote this book is because I couldn't find any books that covered the content in a concise format.
Occasionally the need to review a technology is required, but without having to read an entire book for the parts needed.
I believe that the content of this book is covered in great detail, enough to be useful in every day tasks, as well as preparing a CCIE candidate adequately for the actual lab. This book is a great review guide of the covered topics and the only review guide available to CCIE Candidates.

The earlier draft versions of this book enabled me to pass my CCIE R&S on first attempt. Since then great deal of effort has gone into this book, to ensure that every person reading this, will find it as useful.

I trust you will enjoy reading the Routing-Bits Handbook and hopefully use it as a reference for years to come.

CONVENTIONS

** This book was specifically designed to be printed in landscape mode, to make use of the extra line space for explanations.
** Therefor it is recommended to bind this book on the left side.
** At first this book style might take a bit of time getting used to, but you will quickly understand why this was done.

```
- CONFIG-SETS                          - Are short summarized examples showing how to implement various technologies.
- COMMANDS                             - Lists the command syntaxes with the required and optional strings.
- OUTPUT-101                           - Explains certain command outputs in more detail.
- " " (double quotes)                  - Indicates/refers to a CLI command.
- ' ' (single quotes)                  - Indicates/refers to a command keyword/option.

- Prompt Elements:
      # sh ip route                    - A hash followed by a space, always indicates commands in Privileged EXEC Mode.
      #interface fa0/0                 - A hash without a following space, always indicates commands in Global Configuration Mode.

- Command Elements:
      |     Vertical bars              - Functions as an OR. Line1|Line8.
      []    Square brackets            - Indicates optional strings of a particular command.
      {}    Curly brackets             - Indicates required strings of a particular command.
      (o)   Optional                   - Indicates optional, non-required commands.
```

FEEDBACK

While every effort has been made to ensure this book is complete and as accurate as possible, error and typos are possible.
By letting me know of any mistakes, they can be corrected for the benefit of future releases.

Furthermore I would really appreciate any questions, comments, or feedback sent to <notes@routing-bits.com>.

```
  ______       _            _     _
 / _____)     (_)  _       | |   (_)
( (____  _ _ _ _ _| |_ ____| |__  _ ____   ____
 \____ \| | | | (_   _) ___)  _ \| |  _ \ / _  |
 _____) ) | | | | | |( (___| | | | | | | ( (_| |
(______/ \___/|_|  \__)____)_| |_|_|_| |_|\___ |
                                         (_____|
*-=-=-=-=-=-=-=-=-=-=-=-*
|      INDEX            |
*-=-=-=-=-=-=-=-=-=-=-=-*
```

```
- Layer2 Switchports
      + Speed/Duplex
      + Access Ports
      + Trunk Ports
            o Encapsulation
            o Mode
      + Native VLAN
      + Allowed List
      + 802.1q Tunnel
- VTP (VLAN Trunking Protocol)
      + Domain Name
      + Modes
            o Server
            o Client
            o Transparent
      + Revision Numbers
      + Authentication
      + VTP Pruning
      + Prune Eligible list
      + Backing up vlan.dat
- Layer3 Routing
      + Native Routed Ports
      + SVI (Switched Virtual Interface)
      + Router-on-a-stick
- Ether-channel
      + Dynamic
            o PAgP
            o LACP
      + Static On
      + Layer2 Ether-channels
      + Layer3 Ether-channels
      + Load Balancing
- STP (Spanning-Tree Protocol)
      + Root Bridge Election
      + Root Port Election
      + Path Selection
            o Port Cost
            o Port Priority
      + Timers
      + STP Port Roles and States
      + Advanced Spanning-Tree Features
            o Portfast
            o Uplinkfast
            o Backbonefast
            o BPDU Guard
            o BPDU Filter
            o Root Guard
            o Loop Guard
            o UDLD
      + Disabling STP
- MST (Multiple Spanning Tree)
      + Root Election
      + Path Selection
- RSTP (Rapid Spanning Tree Protocol)
- Advanced Catalyst Features
      + CAM Maintenance
            o Static/Dynamic Entries
            o Aging
            o MAC Notifications
      + SPAN (Switchport Analyzer)
      + RSPAN (Remote Switchport Analyzer)
      + 802.3x Flow-Control
      + Link State Tracking
      + Smartport Macros
      + SDM Templates (Switched Database Manager)
      + Flex Links
      + Private VLANs
- Bridging
      + Transparent
      + CRB (Concurrent Routing and Bridging)
      + IRB (Integrated Routing and Bridging)
      + Fall-Back Bridging
- Security
      + Port Security
            o Violation
            o MAC Addresses
            o Aging
      + 802.1x Authentication
      + Storm Control
      + DHCP Snooping
            o Option-82 Data-Inspection

      + IP Source Guard
      + DAI (Dynamic ARP Inspection)
      + VLAN ACLs
            o IP ACL
            o MAC ACLs & Ethertypes
      + Switchport Protect
      + Switchport Block
```

```
- Troubleshooting Switching
      + Interfaces and Trunks
      + User VLANs and Host Issues
      + VTP
      + Dot1q Tunnels
      + Ether-Channels
      + STP


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
    Layer2 Switchports
*=========================*
- Speed mismatches usually cause a link to change to/from an UP/DOWN status.
- Duplex mismatches will bring the link UP/UP but will typically result in packet loss and interface errors.
 > This is typically seen with the command "sh interface" as 'late collisions'.

- Layer2 Switchport Types
 > Access Ports
      >> Specify which VLAN will carry the traffic for that interface.
      >> Only one VLAN can be configured on the interface.
      >> If none are configured, the interface carries traffic for the default VLAN.
 > Trunk Ports
      >> Can have two or more VLANs configured on the interface.
      >> Can carry traffic for several VLANs simultaneously by encapsulating the frames.
 > Tunnel Interfaces
      >> Are used to build transparent layer2 VPNs.

- Trunk Ports
 > Encapsulation: ISL
      >> Cisco proprietary.
      >> All traffic is encapsulated within a 30-byte ISL frame (26-byte header and 4-byte trailer).
      >> Configured with "sw trunk encapsulation isl".
 > Encapsulation: 802.1q
      >> Open standard.
      >> All traffic has a 4-byte 802.1q tag inserted, except 'native' VLAN traffic, which is sent without this tag.
      >> Dot1q supports the concept of a native VLAN.
      >> Configured with "sw trunk encapsulation dot1q".
 > Mode: Static Trunk
      >> Forces a port to trunking mode.
      >> Configured with "sw mode trunk".
 > Mode: DTP (Dynamic Trunking Protocol)
      >> Is enabled by default.
      >> Default mode depends on the platform:
            >>> 3550 Default mode is dynamic desirable (DD) : Actively initiates the trunk negotiation.
            >>> 3560 Default mode is dynamic auto (DA) : Responds ONLY if trunk negotiation was requested.
      >> To negotiate a trunk, at least one side must be DD or be static 'ON'.
      >> (DD + DD) = Will trunk e.g. ports between 3550 & 3550.
      >> (DD + DA) = Will trunk e.g. ports between 3550 & 3560.
      >> (DA + DA) = Will not trunk by default.
      >> DTP negotiation can only be disabled with "sw nonegotiate".
      >> Setting the interface to static mode with "sw mode access|trunk" will not disable DTP negotiations.
      >> To confirm if DTP is enabled or disabled, use the command "sh int (int) sw | i Nego"
      >> The DTP mode is configured with "sw mode dynamic auto|desirable"
      >> Routers do not support DTP. A switch interface must be manually trunked to a router's trunk interface.

- Native VLAN
 > A trunk port can carry untagged packets simultaneously with the 802.1Q tagged packets.
 > Traffic sent and received on a native VLAN interface does not have an 802.1q tag inserted.
 > The frame is sent as if 802.1q was not configured.
 > When a switch running 802.1q receives a frame with no tag, it is assumed to be part of the native VLAN.
 > Native VLAN ID numbers must match on both ends of the trunk.
 > Default native VLAN is 1.


- Allowed List
 > By default, a trunk port sends traffic to and receives traffic from all VLANs.
 > The allowed-list limits which VLANs are allowed on a specific trunk link.
 > It is in other words VLAN minimization, i.e. when a VLAN is removed from the allowed-list.
 > VLAN-1 is different from other VLANs in that only data traffic is excluded.
      >> Control-plane traffic (CDP,VTP,STP) will still traverse the link using VLAN-1, but data traffic will not.


- 802.1q Tunnel
 > Used to provide transparent layer2 VPN over a switched ethernet network, to carry unicast, broadcast, multicast, CDP, VTP or STP.
 > Uses dot1q inside dot1q to tunnel layer2 traffic.
 > Cannot be dynamically negotiated and traffic is not encrypted.
 > Confirm prior to configuration that underlying end-to-end connectivity is established.
 > When using dot1q tunneling CDP, STP and VTP are NOT carried across the tunnel by default (it must be enabled if needed).
 > Additionally dot1q also supports ether-channels between customer sites.
 > Dot1q-Tunnel requires:
      >> 802.1q trunking end-to-end.
      >> System MTU should be a minimum of 1504 to support the additional 4-byte metro tag.

!!NOTE!! Be careful when running OSPF to a switch with a system MTU of 1504, the adjacency won't come up, this is due to a MTU
      mismatch. Disable the MTU check on the router's OSPF interface with "ip ospf mtu-ignore".

CONFIG-SET: 802.1q Tunnel config on the client facing interface
+-----------------------------------------------------------------
|     system mtu 1504                                   STEP1 - Configures the required MTU size (this requires a restart)
|     !                                                       - Required to allow the extra 4-byte tag
|     vlan dot1q tag native                                   - (o) Allows traffic on the native VLAN to be tagged too
|     !
|     interface fa0/1                                         - The switch interface facing the end point/customer
|      shut                                                   - It's recommended to shut the port before configuring dot1q
|      sw mode dot1q-tunnel                             STEP2 - Enables the dot1q-tunnel on each end-point of the tunnel
|      sw access vlan 515                               STEP3 - This is the switch end-to-end VLAN, i.e. the METRO TAG
|      l2protocol-tunnel {cdp | vtp | stp}                    - (o) CDP: Re-enables CDP for that interface
|      no shut                                                - (o) VTP/STP: Allows the transport of 3rd party layer2 protocols
```

```
-----------
 COMMANDS
-----------
# sh interface status                                    - Shows the interface status, desc, VLAN, duplex, speed, type, etc
# sh interface {int} switchport                          - Shows the layer2 attributes, ie trunk, switchport=enabled/disabled, etc
# sh interface trunk                                     - Shows the trunked interfaces
# sh system mtu                                          - Shows the configured MTU value
# sh l2protocol                                          - Shows the layer2 protocol interfaces and transported protocols


#vlan dot1q tag native                                   - Enables native VLAN traffic to get encapsulated with a dot1q header
#interface range fa0/13 - 21                             - Configures a range of ports
 #sw mode access                                         - Manually sets an interface to access mode
 #sw mode trunk                                          - Manually sets an interface to TRUNK unconditionally
 #sw mode dynamic {auto | desirable}                     - {auto}: Will only respond to DTP trunk negotiation requests
                                                         - {desirable}: Will initiate trunk negotiation using DTP
 #sw nonegotiate                                         - Disables DTP negotiation

 #sw access vlan {vlan}                                  - Assigns a VLAN to an interface
 #sw trunk encapsulation {isl|dot1q}                     - Manually configure the encapsulation mode.
 #sw trunk native vlan {vlan-id}                         - 802.1q : Changes the (def = 1) native VLAN (must match between two switches)

 #sw trunk allowed vlan {all|none|except|remove|add} {vlan ID}
                                                         - Modifies the allowed VLANs on a trunk link
                                                         - {all}: All VLANs are allowed (default)
                                                         - {none}: No VLANs are allowed
                                                         - {add|remove} Add/Remove VLANs to/from the current list
                                                         - {except} Allow all excluding the specified VLANs

#system mtu {mtu}                                        - Configures the required MTU size (requires a restart)
#system mtu routing {mtu}                                - Sets the MTU for routing processes to a different value than system MTU

#interface fa0/1                                         >>> Dot1Q tunnel <<<
 #sw mode dot1q-tunnel                                   - Enables the dot1q-tunnel on customer facing end-point of the tunnel
 #sw access vlan {vlan id}                               - Configures the end-to-end switch VLAN, aka metro-tag
 #l2protocol-tunnel {cdp | vtp | stp}                    - (o) CDP: Enables CDP for that interface (default = disabled)
                                                         - (o) VTP/STP: Allows the 3rd party to attach his layer2 network directly



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
   VTP (VLAN Trunking Protocol)
*====================================*
```

- Is used to advertise VLAN attributes and ease administration.
- Is not a requirement of ethernet networks, because it does not define broadcast domains.
- The VTP domain name is the basic configuration needed for a switch to be part of a domain unless a domain password is configured.

```
- VTP Modes
 > Server (default mode)
      >> Changes are done ONLY on the VTP server.
      >> VLAN configuration is stored in the VLAN database file called vlan.dat and is located on flash (const_nvram).
      >> VLANs 2-1000 are configurable.
 > Client
      >> VTP clients receive their configuration from the VTP server. VTP changes are not allowed on clients.
      >> VLAN configuration is stored in the VLAN database file called vlan.dat and is located on flash (const_nvram).
 > Transparent
      >> Maintains a local database with the VLAN configuration stored in the running config.
      >> Transparent mode is required to configure the extended VLAN range (1006-4096).
      >> VTP updates are sent using the TLV (Type-Length-Value) format.
      >> If the domain name matches the locally configured domain name, a VTP version-2 transparent switch will transparently relay
             transmitted TLV updates between switches, but a VTP version-1 transparent switch will drop those TLV updates.
      >> VLAN add/removes in the VTP domain do not affect transparent switches as the updates are not stored.
      >> A revision of 0 indicates a transparent mode switch is not participating in the update sequence of the VTP domain.

- Revision Numbers
 > Transparent mode will have a revision number of 0, this will not increase with database changes.
 > For every change, a VTP Server's revision number will be increased by 1 and will be propagated to the VTP clients.
 > Higher revision numbers takes preference.
 > If a switch with a matching domain name and a higher revision number connects to the network, its database will be propagated
      to all other switches, potentially overwriting the existing VTP database (regardless whether the switch is configured as
      a VTP server or a VTP client).

- Authentication
 > The domain-name is required to be the same throughout a domain.
 > Even though the passwords are the same, the MD5 hashes could be different. Thus always make sure that the MD5 hashes are the same.
 > Configured with "vtp password {pwd}" and the MD5 hashes are seen with "sh vtp status".

- VTP Pruning
 > Eliminates the need to statically remove VLANs from trunk links where they are not needed, this is done by having the
      switches automatically communicate with each other which VLANs they have locally assigned or are in the transit path for.
 > If a layer2 network is converged, all devices should agree that VTP pruning is enabled, as per "sh vtp status".
 > This reduces broadcast traffic.
 > From the 'show interface pruning':
      >> The field 'VLAN traffic requested of neighbor' indicates what VLANs the local switch told its neighbors, it needs.
      >> The field 'VLANs pruned for lack of request by neighbor' indicates the VLANs that the upstream neighbor did not request.

- Pruning Eligible List
 > Control which VLANs are allowed to be pruned or not, across a link, based on the VLANs assigned locally.
 > Removing a VLAN from the 'prune eligible list' forces the switch to receive traffic for that VLAN.
      Configured with "switchport trunk pruning vlan" command.
 > ONLY VLANs 2-1000 are 'prune eligible', the 5 default VLANs (1, 1002-1005) and extended VLANs cannot be pruned off an interface.

- Backing up vlan.dat
 > Copy the vlan.dat file from const_nvram in flash to either the bootflash partition or to an external TFTP server.
```

```
-----------
 COMMANDS
-----------
# sh interface [int] pruning                         - Shows pruning status
# sh interface trunk                                 - Shows local trunk interfaces, allowed VLANs, etc
# sh vtp status                                      - Shows the VTP configuration. The revision, no of VLANs,
                                                           mode, domain-name, MD5 hash, etc
# sh vtp password                                    - Shows the configured VTP password
# sh vlan brief                                      - Shows the configured VLAN and the associated local interfaces

# copy const_nvram:vlan.dat [bootflash:] [tftp://IP]  - Backs up the vlan.dat file

#vlan 43,156,74,9-25                                 - Creates the specified VLANs
#no vlan 2-1000                                      - Will remove the specified VLANs ranging from 2 to 1000

#vtp mode {server|client|transparent}                - Configures the VTP mode. (default = server)
#vtp password {pwd}                                  - Configures a VTP domain password
#vtp pruning                                         - Enables VTP pruning (must be globally enabled)
#interface fa1/0/1
 #sw trunk pruning vlan 2-8,10-1001                  - VLAN 9 removed from the prune eligible list means
                                                           So traffic for VLAN 9 will be received

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    Layer3 Routing
*=====================*
- Native Routed Ports
 > Same as an ethernet interface on a router.
 > Configured with "no switchport" and "ip address".

- SVI (Switched Virtual Interface)
 > The VLAN must exist in the database, or else the VLAN interface will show as DOWN/DOWN.
 > Configured with "interface vlan {id}".

- Troubleshooting trunking and ports with the help of an SVI.
 > If there are layer2 issues between routers across multiple switches, an easy way to find the problem is:
      >> Create an SVI in that troubled VLAN on one switch at a time.
      >> Assign an IP from the datalink range to the SVI.
      >> Then ping all the routers on that datalink to isolate the problem.
      >> Refer to http://routing-bits/2008/11/05/troubleshooting-vlan-issues/.

- Router-on-a-Stick
 > Is a layer2 enabled switch that is trunked to an external layer3 router, which is used to route between the VLANs.
 > It is the legacy version of a SVI.
 > When configuring, remember that routers do not support DTP.
 > The switch interface must be set to a trunk with "switchport mode trunk".
 > The router encapsulates ISL or 802.1q traffic using sub-interfaces:
      #interface fa0/1.123
       #encapsulation {isl | dot1q} {vlan} [native]
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    Ether-channel
*=====================*
- Ether-channels are independent of the underlying interface mode, i.e. access ports, tunnel ports, trunk ports, or native
      layer3 routed interfaces.
- All member interfaces should have identical configuration.
- ALWAYS SHUT the member interfaces before configuring the ether-channel.
!!NOTE!! When the command "channel-group" is issued, all attributes from the member interfaces are immediately inherited by
      the port-channel interface.

- The mode determines how negotiation occurs
 > On               - No negotiation, forces the channel.
 > Desirable        - Send PAgP initiation messages.
 > Auto             - Listen for PAgP.
 > Active           - Send LACP initiation messages.
 > Passive          - Listen for LACP.

- PAgP (Port Aggregation Protocol)
 > Requires at least one side to be desirable.
 > If both sides are auto, no channel will form.

- LACP (Link Aggregation Control Protocol) also referred to 802.3ad!
 > Requires at least one side to be active.
 > If both sides are passive, no channel will form.

- PAgP and LACP are not compatible; both ends of a ether-channel must use the same protocol.
- The "channel-protocol" command is used to lock the mode from being changed undesirably when using the "channel-group mode".

- Layer2 Ether-Channel
 > A successful layer2 ether-channel will show (SU) with the command "sh ether-channel-channel summary".
 > A unsuccessful layer 2 ether-channel will show (SD).


CONFIG-SET: Layer2 Ether-Channel
+-------------------------------------
|     interface range fa0/20-22
|      shutdown                                                 - Shuts the physical interfaces
|      switchport trunk encapsulation isl
|      switchport mode trunk                                    - Changes the interface mode to a trunk
|      channel-group 34 mode desirable                          - Specifies the interface part of an ether-channel using PAgP
|     !
|     interface port-channel 34
|      sw trunk encap isl                                       - Configures the layer2 channel parameters (default = inherited)
|      sw mode trunk                                            - Best practice to configure again
|     !
|     interface range fa0/20-22
|      no shutdown                                              - Bring the member interfaces and the port-channel up
```

```
- Layer3 Ether-channel
 > Issue the "no switchport" command on all the member interfaces to enable it as a layer3 interface.
 > Successful layer3 ether-channels will show (RU) with the command "sh ether-channel summary".
 > A unsuccessful layer 3 ether-channel will show (RD).


CONFIG-SET: Layer3 Ether-Channel
+-------------------------------------
|      interface range fa0/15-18
|       shutdown                                        - Shuts the physical interfaces
|       no switchport                                   - This will enable layer3 channel on the interfaces
|       channel-group 12 mode active                    - Configures the ether-channel with the protocol: LACP (802.3ad)
|      !
|      interface port-channel 12
|       ip address 10.10.10.1 255.255.255.0             - Configures an IP address on the layer3 ether-channel
|      !
|      interface range fa0/15-18
|       no shutdown                                     - Bring the member interfaces and the port-channel up

- Ether-channel Load-Balancing options are configured with "port-channel load-balance {mode}":
 > dst-ip          - Destination IP Address.
 > dst-mac         - Destination MAC Address.
 > src-dst-ip      - Source XOR Destination IP Address.
 > src-dst-mac     - Source XOR Destination MAC Address.
 > src-ip          - Source IP Address.
 > src-mac         - Source MAC Address (Default).


-----------
 COMMANDS
-----------
# sh ether-channel summary                      - Shows the port-channel status and interfaces
# sh ether-channel load-balance                 - Shows the load-balancing configuration mode
# sh ether-channel {id} port-channel            - Shows port-channel specific information
# sh spanning-tree vlan {vlan id}               - Used to verify layer2 an ether-channel
                                                - If member interfaces are in FWD mode, then a channel is broken
# sh interfaces trunk                           - Used to verify layer2 an ether-channel
                                                - Member interface should not be seen as trunks
# sh ip route                                   - Used to verify layer3 an ether-channel
                                                - The port-channel interfaces should be installed not the member interfaces
# sh lacp sys-id                                - Shows the Dot1q LACP system priority

#lacp system-priority {priority}                - Sets LACP system-priority (lower priority is preferred)
#port-channel load-balance {lb mode}            - Configures the load-balancing mode (see options above)
#interface range fa0/15-18
 #channel-group {no} mode {channel mode}        - Configures the ether-channel, specify the channeling protocol
 #channel-protocol {lacp|pagp}                  - (o) Enforce the channeling protocol used
```

STP (Spanning-Tree Protocol)

```
- STP is used to prevent layer2 bridging loops.
- PvST (Per-VLAN Spanning Tree) maintains a spanning tree instance for each VLAN configured in the network.
- PvST is Cisco proprietary, enabled by default and Cisco's default implementation of STP.

- BPDU (Bridge Protocol Data Unit)
 > Is a packet used to advertise spanning-tree protocol information.

- The STP Root Bridge is elected based on the LOWEST BID (Bridge ID).
- The BID consists of:
 > Bridge priority - consisting of
      >> Priority (default = 32768) (configured in increments of 4096).
      >> Sys-id-ext = VLAN.
 > MAC address.

- The switch that gets elected the root bridge:
 > Will show 'this bridge is root' from "sh span vlan".
 > Will show the same priority and MAC address for both the Root ID and Bridge ID.
 > Will have all its interface for that VLAN in designated forwarding state.

- Root Port Election (upstream port closest to root bridge) is based on:
 1st> Lowest cumulative cost to the root:
      >> Inverse value based on interface bandwidth (an interface with a higher bandwidth will have a lower cost).
 2nd> Lowest upstream BID:
      >> Used to isolate multiple connections to the same upstream bridge.
 3rd> Lowest port ID:
      >> Lowest port priority (0-255) (default = 128)
      >> Lowest port number, i.e. Fa0/5 = 5.

- Influencing the Root Port Election
 > Port Cost
      >> Can be changed to influence how the local switch elects its local ROOT port upstream.
      >> Changing the port cost will affect all the downstream switches, as cost is the sum of all the port costs to the root.

 > Port Priority
      >> Can be changed to influence how a downstream switch elects its root port.
      >> Priority is locally significant between two directly connected switches.
      >> Upstream port priority can be seen with "sh span vlan {id} detail"  as 'designated port id x.x'.

- Timers
 > Downstream devices from the root bridge inherit the timers configured on the root.
 > Default timers and their purpose are:
      >> Hello Time (2 sec.)         - Determines how often the switch broadcasts its hello BPDUs to other switches.
      >> Max Age (20 sec.)           - Age limit when outdated received protocol information is discarded.
      >> Forward Delay (15 sec.)     - The time spent by a port in each of the learning and listening states.
```

```
- STP Port Roles
 > Root port
      >> Is the one port on a switch that is closest (with the lowest root path cost) to the root bridge.
 > Designated port
      >> Is the downstream port on a LAN segment that is closest to the root. This port relays, or transmits BPDUs down the tree.
 > Blocking port
      >> Is a port that is neither root nor a designated port.
 > Alternate port
      >> Is a port that is a candidate root port in the blocking state (next-closest to the root bridge).
      >> These ports are identified for quick use by the STP uplinkfast feature.
 > Forwarding port
      >> Ports where no other STP activity is detected or expected. These are ports with normal end-user connections.

!!NOTE!! MAC addresses should only be learned on root or designated ports.

- STP Port States
 > Disabled
      >> Ports that are in a DOWN state.
      >> This state is special and is not part of the normal STP progression for a port.

 > Blocking
      >> ONLY when a port initializes will it be in the blocking state.
      >> The port is allowed to receive only BPDUs so that the switch can hear from other neighboring switches.
      >> The port cannot receive or transmit data and cannot add MAC addresses to its address table.
      >> Blocking delay = 20 sec. (this value CANNOT be changed).
 > Listening
      >> A port is moved from blocking state if the switch thinks that the port can be selected as a root port or designated port.
      >> The port is allowed to receive and send BPDUs so that it can actively participate in STP.
      >> The port still cannot send or receive data frames.
      >> Listening delay = 15 sec.

 > Learning
      >> After the listening delay, the port is allowed to move into the learning state.
      >> The port still sends and receives BPDUs as before.
      >> The switch can now learn new MAC addresses to add to its address table.
      >> The port cannot yet send any data frames.
      >> Learning delay = 15 sec.

> Forwarding
      >> After the forward delay (listening and learning states) (default = 30 sec) the port transitions to the forwarding state.
      >> The port now can send and receive data frames, collect MAC addresses in its address table and send and receive BPDUs.

- Important things to know about port states:
 > RFC dictates that Listening and Learning times must be equal values.
 > Blocking state delay ONLY applies when a port first initializes, i.e. after a reboot, not when a port transitions to forwarding.
 > When a port transitions to the forwarding state, the delay is only the listening and forwarding delay, i.e. unshutting a port.
 > So when a port first comes up there is a collective delay of 50 sec. (20+15+15) of no data flow.
 > And when a port changes state the collective delay is only 30 sec. (15+15) of no data flow.
 > Be careful how questions could be phrased with regards to the delay times.
```

- Portfast

> Is used to bypass the forwarding delay thus a port transitions immediately to the forwarding state.
>
> Enabling portfast on a non-host port could create loops.
>
> Configured globally with "spanning-tree portfast default".
>
> Interface configuration "spanning-tree portfast enable".

- Uplinkfast

> Cisco proprietary.
>
> Is used to speed up convergence time when direct failure of the local root port occurs.
>
> When a root port fails, the next alternate port is immediately transitioned to the root port and placed into the forwarding state.
>
> The CAM table is flooded out of this new root port to expedite the learning phase of its upstream neighbors.
>
> Configured globally with "spanning-tree uplinkfast".

- Backbonefast

> Cisco proprietary
>
> Used to speed up convergence when an indirect failure occurs upstream in the network by immediately expiring the 'MAX_AGE' timer.
>
> Generates RLQ (Root Link Query) PDUs to check if the switch should expire the 'MAX_AGE' for its current BDPUs and begin convergence.
>
> Configured globally with "spanning-tree backbonefast".

- BPDU Guard

> Used to enforce access layer security, when an erroneous BPDU is received on an access interface, by transitioning the interface to shutdown and ERR-DISABLE state.
>
> Err-disable recovery can be configured to automatically bring the interface out of ERR-DISABLE state after the configured interval.
>
> The ERR-DISABLE state can be seen with "sh interface status".
>
> Configured globally with "spanning-tree portfast bpduguard default".
>
> Interface configuration "spanning-tree bpduguard enable".

- BPDU Filter

> Discards all inbound BDPUs and does not send BDPUs out of the interface.
>
> Unlike BPDU guard, the interface does not go into ERR-DISABLE state when a violation occurs.
>
> Other user traffic will still be forwarded.
>
> If BPDU filter default is enabled with portfast, all the interfaces will run in portfast mode except those that are receiving BPDUs.
>
> Configured globally with "spanning-tree portfast bpdufilter default".
>
> Interface configuration "spanning-tree bpdufilter enable".

- Root Guard

> Similar to BDPU guard, the difference is that a root guard interface is only disabled if a superior BPDU was received, placing the interface into a 'ROOT_INCONSISTANT_STATE'.
>
> It should be enabled on a downstream interface, an interface which should never become a root-port.
>
> A superior BPDU indicates a better cost to the root bridge than what is currently installed.
>
> Interface configuration "spanning-tree guard root".

- Loop Guard

> Is used to prevent STP loops from occurring due to a unidirectional link.
>
> Similar to UDLD but instead uses BDPU keepalive to determine unidirectional traffic.
>
> If a blocked port transitions to the forwarding state erroneously, a loop can occur.
>
> Blocked ports will be transitioned into a 'LOOP_INCONSISTANT_STATE' to avoid loops.

```
 > Interface configuration "spanning-tree guard loop".

- UDLD (Unidirectional Link Detection)
 > Cisco proprietary.
 > Uses its own keepalives to prevent loops, by detecting a failure on the TX ring, but not the RX ring.
 > This is why UDLD has to be configured on both sides of a link.
 > UDLD is typically used with fibre optic cables.
 > Peers discover each other by exchanging frames sent to the MAC-address 0100:0CCC:CCCC.
 > The global command "udld enable" only applies to fibre interfaces!!!
 > The interface command "udld port [aggressive]" applies to all other interfaces.
 > To enable UDLD for copper interfaces, use the interface command "udld port aggressive"
 > Two modes:
      >> Normal         - Informational mode, generates a log entry, but doesn't disable or shutdown the port.
      >> Aggressive     - Will place an interface into the ERR-DISABLE state.


- To test BPDU filters from the router connecting to a switch, configure the following on the router:
      #bridge 1 protocol ieee
      #interface eth0
       #bridge-group 1

- Disabling STP
 > STP cannot be directly disabled on a per interface basis, but can be disabled on a per-VLAN basis or globally on the switch.
 > Configured with the "no spanning-tree vlan vlan-id" command.
 > Alternatively the BPDUs on an interface could be filtered to create the same effect as 'disabling' STP on the interface.
 > FLEX-links also disables STP on an interface, but use it cautiously.


 ----------
 COMMANDS
 ----------
# sh spanning-tree summary                          - Shows the STP mode, summary of all VLAN timers
# sh spanning-tree root                             - Shows the status and configuration of the root bridge
# sh spanning-tree [vlan {id}] [detail]             - Shows the root bridge, the local Root ID and Bridge ID
                                                    - Shows the root/designated/alternate ports
                                                    - [detail] Will show more information per interface per VLAN
# sh spanning-tree interface {int} portfast         - Shows if portfast is enabled or not
# sh errdisable recovery                            - Shows which ERR-DISABLE reasons are enabled
# sh udld {interface}                               - Shows the UDLD state and counters

# debug spanning-tree events                        - Nice debug to see port state changes

#spanning-tree mode {pvst | rapid-pvst | mst}       - Configures the spanning-tree mode (default = PVST)
#spanning-tree vlan {id/s} priority {value}         - Manually set the bridge priority (default = (32768 + SYS-ID-EXT)
                                                    - {value}: Must be increments of 4096 (lowest value is preferred)
#spanning-tree vlan {id/s} root {primary | secondary} [diameter {2-7}]
                                                    - {primary}: Configures a priority of 24576, if not low enough, 4096 is used
                                                    - {secondary}: Configures a priority of 28672
                                                    - [diameter]: Maximum number of switches between any two points
```

```
#no spanning-tree extend system-id                  - Disables SYS-ID-EXT. (default = enabled) (PVST & Rapid PVST only)
#spanning-tree vlan {id/s} hello-time               - Sets the hello interval (default = 2sec)
#spanning-tree vlan {id/s} forward-time             - Sets the forward delay (default = 15sec for each delay)
#spanning-tree vlan {id/s} max-age                  - Sets the max age interval (default = 20sec)

#spanning-tree portfast default                     - Enables portfast globally on all access ports
#spanning-tree portfast bpduguard default           - Enables portfast BPDU guard on all access ports
#spanning-tree portfast bpdufilter default          - Enables portfast BPDU filter globally
#spanning-tree uplinkfast                           - Enables the uplinkfast feature
#spanning-tree backbonefast                         - Enables the backbonefast feature
#spanning-tree loopguard default                    - Enables loopguard on all ports

#udld enable                                        - Enables UDLD protocol on all fibre interfaces
#errdisable recovery cause [bpduguard]              - Allow different causes to be recovered, after the time specified
#errdisable recovery interval {sec}                 - Time to pass before recovering from ERR-DISABLE state
                                                    - Changes the (default = 300sec) ERR-DISABLE recovery timer


#interface fa0/2
 #spanning-tree [vlan] cost {value}                 - Adjusts the port cost manually for all or single VLAN
                                                    - Lowest value is preferred
 #spanning-tree [vlan] port-priority {value}        - Adjusts the port priority in increments of 16 (default = 128)
 #spanning-tree bpdufilter {enable | disable}       - En/Disables not sending/receiving BPDUs on an interface. Silently discarded
 #spanning-tree bpduguard {enable | disable}        - En/Disables accepting BPDUs on the interface (violation = err_disable)
 #spanning-tree portfast {enable|disable} [trunk]   - Enables portfast and optionally even if in trunk mode
 #spanning-tree guard root                          - Enables STP root guard for all VLANs on the interface
 #spanning-tree guard loop                          - Enables STP loop guard for all VLANs on the interface
 #spanning-tree guard none                          - Disables guard filters on the interface
 #udld port [aggressive]                            - Enables UDLD for copper interfaces, optionally as aggressive

#no spanning-tree vlan {vlan-id}                    - Disables STP per-VLAN
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
    MST (Multiple Spanning Tree)
*==================================*
```

- IEEE standard defined in 802.1s.
- Allows user-defined STP instances to be mapped to multiple VLANs.
- If no instances are defined, all VLANs are mapped to instance 0.
- Same election process as STP. MST also uses the lowest BID in the network to elect the Root Bridge.
- With MST there is only one election per user-defined instance.
- MST also uses a cost value derived from the inverse bandwidth of the interface.
- When MST is enabled, RSTP is automatically enabled.

```
-----------
 COMMANDS
-----------
# sh spanning-tree mst [instance number] [detail]   - Shows the MST root bridge, local root/bridge ID, port states
                                                    - [detail] Shows more information per interface per VLAN
```

```
#spanning-tree mode mst                                - Configures the STP mode to MST
#spanning-tree mst configuration                       - Enters MST config sub-mode
 #name MST1                                            - Sets configuration name
 #revision 1                                           - Sets configuration revision number
 #instance 1 vlan 1-200                                - Assigns VLANs 1-200 to instance 1
 #instance 2 vlan 201-4094                             - Assigns VLANs 201-4094 of the VLANs to instance 2
#spanning-tree mst 1 priority 0                        - Sets the bridge priority for the spanning tree instance 1 to zero


#interface fa0/4
 #spanning-tree mst {instance} cost {value}            - Changes the interface spanning tree path cost for an instance
 #spanning-tree mst {instance} port-priority (value)  - Changes the spanning tree port priority for an instance (multiples of 16)


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================================*
   RSTP (Rapid Spanning Tree Protocol)
*=======================================*
- IEEE standard defined 802.1w.
- Designed to speed up STP convergence through a reliable handshaking process.
- RSTP Port Roles
 > Root port
      >> Is the port that has the best root path cost to the root.
 > Designated port
      >> Is the downstream port that has the best root path cost to the root.
      >> Is a downstream interface pointing away from the root bridge.
 > Alternate port
      >> Is a port that has an alternate path to the root. An alternate port is less desirable than the root port.
      >> In the blocking state the port will receive STP info, but not send any out the interface.
 > Backup port
      >> Is a backup designated port.

- RSTP Port States
 > Discarding
      >> Incoming frames are simply dropped; no MAC addresses are learned.
      >> Combines the 802.1D (STP) disabled, blocking and listening states.
 > Learning
      >> Incoming frames are dropped, but MAC addresses are learned.
 > Forwarding
      >> Incoming frames are forwarded according to MAC addresses that have been (and are being) learned.

- RSTP Fast Transitioning
 > Works only on point-to-point links between two switches.
 > By default, a switch derives the link type of a port from the duplex mode.
 > A full-duplex port is considered a point-to-point link while a half-duplex link is assumed to be a shared link.
 > If the port is designated as a shared link, RSTP fast transition is forbidden, regardless of the duplex setting.
 > Configured with "span link-type".
```

```
-----------
 COMMANDS
-----------
# sh spanning-tree interface                               - Shows information about the spanning-tree state

#spanning-tree mode rapid-pvst                             - Enables Rapid PvST+ mode
#interface fa0/1
 #spanning-tree link-type {shared | point-to-point}        - Specifies the link type for RSTP fast transition



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*================================*
    Advanced Catalyst Features
*================================*
- CAM Maintenance
 > Static Entries
      >> Could be useful to statically hard-code which MAC addresses are reachable via which ports.
      >> Another use is to Null-switch a MAC address. If the interface is down, traffic to that MAC will still be dropped.
      >> Static MAC entries always override dynamically learned MAC entries.
 > Dynamic Entries
      >> MAC addresses are recorded based on the interfaces they were received on.
 > Aging
      >> MAC addresses are aged out if no update is received.
      >> By default the aging time is 300 sec. if not configured explicitly.
 > MAC notifications
      >> SNMP could be used to log when a MAC address is added or removed from the CAM table.
      >> MAC notifications are generated only for dynamic and secure MAC addresses.


- SPAN (Switchport Analyzer)
 > Is used to redirect traffic from a port or VLAN onto another for analysis by devices such as a packet sniffer or IPS.
 > By default traffic coming in on the destination SPAN port will get dropped.
 > The [ingress] keyword tells the switch which access VLAN inbound traffic on the destination port should belong to.
 > Copy descretly owned by Kane  Bagwell

- RSPAN (Remote Switchport Analyzer)
 > Feature is used when the source port or VLAN that is being monitored is on a different physical switch than the sniffer.
 > First step is to configure the RSPAN VLAN, which carries special attributes.
 > Next, configure the source of the traffic for the SPAN session and direct it to the RSPAN VLAN.
 > Lastly on the switch with the attached sniffer, create a SPAN session with the source as the RSPAN VLAN and the destination as
      the port the sniffer is attached.

- IEEE 802.3x Flow-Control
      > DOC-CD LOCATION
       > Switches, LAN Switches, Config Guides
        > Catalyst 3560 Switch Software Config Guide, Rel. 12.2(25)SEE
         > Configuring Interface Characteristics
          > Configuring IEEE 802.3x Flow Control
 > Flow-control is a mechanism that allows the receiving party of a connection to control the rate of the sending party.
 > A station on a point-to-point link will send a special 'PAUSE' frame to signal the other end of the connection to pause
       transmission for a certain amount of time (the amount is specified in the frame).
 > The PAUSE frame is sent to a reserved multicast MAC address 01:80:C2:00:00:01 using MAC LLC encapsulation.
 > Flow-control is a legacy technology, used to control the sending rate of a host. Newer MLS QOS technologies are more evolved.
 > It is recommended to turn 802.3X flow control off when MLS QOS is enabled.
 > Catalyst 3560 ports can receive, but not send, PAUSE frames.
 > By default flow-control is disabled and you can only enable a Cisco switch to receive PAUSE frames, but not to send them.
 > Configured with "flowcontrol receive on" under an interface.

- Link-State Tracking
 > Link-state tracking, also known as trunk failover, is a feature that binds the link state of multiple interfaces.
 > It is configured in a primary or secondary relationship known as teaming. If the link is lost on the primary interface,
       connectivity is transparently changed to the secondary interface.

- Smartport Macros
 > Used to define a well known template of config to apply onto multiple interfaces.
 > There are default macros on a switch, these can be seen with "sh parser macro [brief]".
 > To apply a default macro use "macro apply {name} {options}".

- SDM Templates (Switched Database Manager)
 > SDM is used to alter the default allocation of resources (i.e. unicast routes, MAC addresses, etc.).
 > By default the 3560 will support 8000 unicast routes,(6000 directly connected, 2000 non-directly connected).
 > Changing the SDM template requires a restart for the changes to take effect.
 > The SDM template and the switch MTU values are stored in flash in a file called 'env_vars', which can be seen
      with the "more" command.
 > Because this info is stored in flash, it will survive a reboot and config erase.

- Flex Links
 > Used as an alternative to STP in environments where physical loops occur in the layer2 network.
 > Works similarly to the "backup interface", whereby there is an 'ACTIVE' link and a 'BACKUP' link.
 > The backup link operates in standby mode, waiting for the line protocol on active link to go down, before coming up.
 > When the active link comes back up, the backup link goes back to standby.
 > STP is automatically disabled on both link types when Flex Links are enabled.

- Private VLANs
 > Can split a single broadcast domain, defined by a single VLAN, into multiple isolated broadcast subdomains
      as defined by a primary VLAN and secondary VLANs.
 > Basically there are VLANs inside a VLAN.
 > Commonly used in shared layer2 environments, like ISP co-locations/hotel rooms, so two sites/rooms can't communicate directly.
 > PVLANs can only be configured when a switch is in VTP transparent mode!!!
 > PVLAN are different from Switchport Protection; PVLANs can span multiple switches whereas Protected ports can't.
 > Private VLAN information is NOT propagated via VTP.
 > Secondary VLANs (isolated and community) do not run their own instance of spanning-tree.

 > Defining the different port roles:
      >> Promiscuous ports      - Are allowed to talk to all other ports within the VLAN.
                                - Are the roles assigned to the primary VLAN ports.
      >> Community ports        - Are allowed to talk to any other port in the same community.
      >> Isolated ports         - Can only talk to other promiscuous ports.
 > Configuring:
      1- Create the secondary VLANs as community or isolated.
      2- Create the primary VLANs and associate the secondary VLANs.
      3- Assign ports to the primary VLAN and secondary VLANS.
      4- Define the association. This limits which other ports the local port can communicate with.


-----------
 COMMANDS
-----------
# sh mac-address-table [static|dynamic] [int][vlan]   - Shows the CAM table
# sh mac-address-table aging-time                     - Shows the MAC address aging time
# sh mac-address-table notification                   - Shows the MAC address notification settings on all interfaces
# sh monitor session {session no}                     - Shows the SPAN configuration
# sh parser macro [brief]                             - Shows the configured macros, as well as the default macros
# sh sdm prefer                                       - Shows the current SDM template

# clear mac address-table notification                - Clears the MAC address notification global counters
# debug back all                                      - Enables debugging for the backup interface

#mac-address-table static {mac} vlan {id} int         - Hardcodes a MAC address to an interface
#mac-address-table static {mac} vlan {id} drop        - Null-switches a MAC address
#mac-address-table aging-time {sec}                   - Configures the maximum aging time for entries in the layer2 table
#mac address-table notification [history|interval]   - Enables the MAC address notification feature
#snmp-server enable traps                             - Sends the SNMP MAC notification traps
#snmp trap mac-notification                           - Enables the SNMP MAC notification trap on a specific interface


#monitor session 1 source {int | vlan}                - Specifies the local source interface of the traffic to span
#monitor session 1 dest int {int} [encap | ingress]   - Setup SPAN to the destination interface
                                                      - [ingress]: Associates inbound traffic on the SPAN port to a VLAN

#vlan 200                                             >>> RSPAN Example <<<
 #remote-span                                         - Enables VLAN 200 to be a RSPAN VLAN
#monitor session 1 source interface fa0/2 [tx|rx|both]- Specifies the source interface to RSPAN and the direction (Def=BOTH)
#monitor session 1 destination remote vlan 200        - Fa0/2 received traffic will be redirected to the RSPAN VLAN-200
#monitor session 1 source remote vlan 200             - Configures another switch to source the RSPAN VLAN-200 traffic
#monitor session 1 dest int fa0/24 ingress vlan 146   - RSPAN traffic is redirected to the host connected to fa0/24
                                                      - Inbound traffic to be places in VLAN-146

#interface fa0/2                                      >>> Flow-Control <<<
 #flowcontrol {receive} {on | off | desired}          - {desired}: Enables flow-control if a host requested it (Default = off)

#interface fa0/3
 #sw voice vlan {id}                                  - Tells the IP-phone which VLAN to be used for voice traffic
 #mls qos trust device cisco-phone                    - Determines if frames with a COS are maintained instead of remarked

#link state track {number}                            >>> Link-State Tracking <<<
#interface range fa0/20-22                            - Enabled by creating the group (1-10)
 #link state group {number} {upstream|downstream}     - Configures the interface as either an upstream or downstream interface
```

```
#macro name {name}                                    >>>  Switchport Macros<<<
 switchport mode access                               - This custom macro can configure multiple interface
 switchport access vlan 146                           - By using a #, the line will act as description
 spanning bpdufilter enable
#interface range fa0/10-13
 #macro apply {name}                                  - Applies the macro to set of interfaces
#interface fa0/9
 #marco apply cisco-default $access-vlan 10           - Applies a default macro and specifies the required options field to VLAN-10


#sdm prefer {routing|vlan|access|dual-ipv4-and-ipv6|default}
                                                      - Changes the SDM-template. Requires a restart to take effect

#interface fa0/4                                      >>> FLEX Links <<<
 #sw backup int fa0/5                                 - Enables fa0/5 as the backup interface to fa0/4
 #sw backup int fa0/5 preemption mode {bw | forced}   - Enables preemption either on higher bandwidth else on interface status
 #sw backup int fa0/5 preemption delay 20             - Time to wait before the preemption kicks in

#vlan 10                                              >>> Private VLANs <<<
 #private-vlan community                        STEP1 - Configures the secondary VLAN as a community private VLAN
#vlan 20
 #private-vlan isolated                         STEP1 - Configures the secondary VLAN as an isolated private VLAN
#vlan 1
 #private-vlan primary                          STEP2 - Configures the VLAN as a primary private VLAN
 #private-vlan association 10,20,30             STEP2 - Configures association between private VLANs
#interface fa0/6
 #sw mode private-vlan promiscuous              STEP3 - Sets the port mode to private VLAN promiscuous
 #sw private-vlan mapping 1 10,20,30            STEP4 - This port is promiscuous in VLAN-1 and can talk to ports in VLAN-10,20,30
#interface fa0/7
 #sw mode private-vlan host                     STEP3 - Sets port mode to private VLAN either isolated/community based on VLAN
 #sw private-vlan host-association 1 10         STEP4 - Member of private VLAN-1 and secondary VLAN-10 can talk to
                                                            any ports in VLAN-10


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    Bridging
*=====================*
- DOC-CD LOCATION
       > 12.4 Mainline Config Guides
        > IBM Technologies
         > Bridging and IBM Networking Config Guide
          > Part 1: Bridging

- IOS can route or bridge a protocol, not both. Defaults:
 > Routers have IP routed.
 > Switches have IP bridged.
```

- Transparent bridging is subject to normal STP rules.
 > Only one active path.
 > Root bridge election.
 > Root port election.

- IRB and CRB are useful when the broadcast domain for one protocol must be extended while maintaining it for another protocol.
- Routers running in bridged mode don't support the 'SYS-ID-EXT', so the bridge priority will be 32768 only, for any VLAN.
- With a lower bridge priority on the router, the router will be elected the root of the spanning tree over a switch.

- CRB (Concurrent Routing and Bridging)
 > With CRB a protocol can be routed on one interface while being bridged on another interface.
 > When CRB is used traffic in the routed domain cannot be passed onto the bridge domain.
 > CRB is considered legacy since IRB includes all the functionality of CRB with the addition of the BVI.

- IRB (Integrated Routing and Bridging)
 > With IRB a protocol can be both routed and bridged on the same interface.
 > When IRB is used traffic from the routed domain can be passed onto the bridge domain.

```
CONFIG-SET: IRB (Integrated Routing and Bridging)
+----------------------------------------------------
|     bridge 1 protocol ieee                         - Creates transparent bridge group
|     !
|     bridge irb                                     - Enables IRB to define the bridged protocol
|     bridge 1 route ip                              - Enables routing and bridging (default)
|     !
|     interface fa0/0
|      bridge-group 1                                - Enables the bridge-group member interfaces
|     !
|     interface bvi 1                                - Configures BVI to connect the bridged and routed domain
|      ip add 10.5.0.1 255.255.255.0
```

- FallBack Bridging
 > Aka VLAN bridging
 > Its main use is to allow machines that speak non-routed or non-supported protocols (SNA, DECNet, AppleTalk, etc.)
    to communicate across VLANs and routed ports.

```
CONFIG-SET: FallBack Bridging
+-------------------------------------------
|     bridge 1 protocol vlan-bridge                  - Specifies the bridging VLAN
|     !
|     interface vlan1                                - Assigns the SVI and routed port to bridge-1
|      bridge-group 1
|     interface fa0/1
|      no switchport
|      bridge-group 1
|     !
```

```
-----------
 COMMANDS
-----------
# sh interface irb                                    - Shows the IRB configuration and interfaces
# sh bridge {group number}                            - Shows the equivalent of a CAM table
# sh spanning-tree                                    - Shows the STP information on a router

#no ip routing                                        - Disables IP routing
#bridge 1 protocol ieee                               - Configures transparent bridge group. This initiates the STP process
#bridge irb                                           - Enables IRB
#bridge 1 bridge ip                                   - Enables bridging for the bridge-group (default)
#bridge 1 route ip                                    - Enables routing and bridging for the bridge-group

#interface fa0/0
 #bridge-group 1                                      - Applies the bridge group to the interface
#interface bvi 1                                      - Configures BVI to connect the bridged and routed domain
 #ip add 1.2.3.4 255.255.255.0                        - Layer3 options applied to the BVI

#bridge 2 protocol vlan-bridge                        - Enables the fallback bridge group
#interface vlan 2
 #bridge-group 2                                      - Applies the bridge-group to SVI or routed interface


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   Security
*======================*
- Port Security
 > Is used to limit access to a port based on MAC addresses.
 > Can only be configured on static access or trunk ports. No dynamic links.
 > By default, once a port goes into ERR-DISABLE it doesn't come out unless:
      >> Shut/no shut
      >> ERR-DISABLE recovery configured (see below)

 > A security port cannot be a destination port for SPAN nor belong to an ether-channel nor be a private-VLAN port.
      It can be configured, but it won't work.
 > Occasionally when port-security is configured with two secure MAC addresses, the port might still go ERR-DISABLE. Just increase
      the allowed amount to three MAC addresses.
!!NOTE!! Remember if HSRP is configured on the security port, the HSRP MAC address must be explicitly allowed.

 > Violation mode
      >> Shutdown mode (Default mode)
            >>> Will disable the port by placing it in the ERR-DISABLE state.
            >>> Will generate a SNMP trap and a syslog message.
      >> Protect mode
            >>> Will not accept traffic from new devices once violation occurs.
            >>> This mode disables learning when any VLAN reaches the max limit and is not recommended on trunk ports.
      >> Restrict
            >>> Will not accept traffic from new devices once violation occurs.
            >>> Will generate a SNMP trap and a syslog message.
```

- 802.1x Authentication
  > Used for username/password authentication between a client and a switch.
  > DO NOT forget to add "aaa authentication login default none", otherwise you might lock the switch and forfeit any points related to that switch.
  > Use AAA with RADIUS for authentication.
  >> Configured with "aaa authentication dot1x"

- Storm Control
  > Limit the amount of unicast/broadcast/multicast traffic accepted on a port.
  > Traffic above the multicast rate suppresses unicast, broadcast and multicast traffic.
  > With storm control it is recommended to hardcode the interface speed to get around the 10/100/1000 negotiation issue.
  > Configured with "storm-control {broadcast | multicast | unicast}"
  > An interface threshold level of 100 means that there is no cap, while an interface threshold level of 0 means no traffic of the designated type is allowed.

- DHCP Snooping
  > DHCP snooping is a feature that provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding database.
  > DHCP snooping acts like a firewall between untrusted hosts and DHCP servers.
  > One can use DHCP snooping to differentiate between untrusted interfaces connected to the end user and trusted interfaces connected to the DHCP server or another switch.

  > Option-82 Data Inspection
  >> A subscriber device is identified by the switch port through which it connects to the network (in addition to the MAC).
  >> Enabled by default when DHCP snooping is enabled globally.
  >> If the DHCP relay agent is enabled but DHCP snooping is disabled, the DHCP option-82 feature is not enabled.

```
CONFIG-SET: DHCP Snooping on a switch
+----------------------------------------------------------
|Configured on SW1 that is connected to VLAN-17 where the DHCP server (R1) is connected
|
|SW1# ip dhcp snooping                                   - Enables DHCP snooping globally
|     ip dhcp snooping vlan 17                           - Enables for VLAN-17
|     !
|     no ip dhcp snooping information option             - Allows R1 to accepts inspected DHCP packets, forwarded from SW1
|     !                                                  - i.e. option-51 (refer to Services chapter for DHCP options)
|     interface FastEthernet 0/1
|      ip dhcp snooping trust                            - Allows R1 to act as DHCP (R1 connected on fa0/1)
|      ip dhcp snooping limit rate 100                   - Limits DHCP messages from R1 to 100 packets/sec
```

- IP Source Guard
  > IP source guard is a security feature that restricts IP traffic on non-routed layer2 interfaces by filtering traffic based on the DHCP snooping binding database and on manually configured IP source bindings.
  > IP source guard is supported only on layer2 ports, which include access ports and trunk ports.
  > IP source guard can be configured with just source IP address filtering(default) or with source IP and MAC address filtering.
  > It requires DHCP snooping to be enabled, otherwise the filtering might not work properly.
  > By default, IP source guard is disabled.
  > Configured with "ip verify source".

- DAI (Dynamic ARP Inspection)

> Helps prevent malicious attacks on the switch by not relaying invalid ARP requests and responses to other ports in the same VLAN.
> Dynamic ARP inspection associates a trust state with each interface on the switch.
> Packets arriving on trusted interfaces bypass all dynamic ARP inspection validation checks and those arriving on untrusted interfaces undergo the dynamic ARP inspection validation process.
> By default, all interfaces are untrusted.

- VLAN ACLs

> Used to apply a layer3 filter to layer2 transit traffic.
> Uses route-map logic to permit(forward) or deny(drop) traffic.
> Changes made to the access-map, will not take effect until the access-map is removed and re-applied.
> ONLY an ACL-permit performs the "forward"/"drop" function in the access-map. An ACL-deny will be ignored. So to deny traffic with VLAN ACLs, permit the traffic and use a "drop" action in the access-map.
> MAC-ACLs will only match NON-IP traffic.
> The Cisco 3560 switch sees IPv6 traffic as IP-traffic, but a Cisco 3550 switch sees IPv6 traffic as NON-IP-traffic.
> Ethertypes are not fully listed on IOS command help or DOC-CD, so memorize them!

```
    >> 0x0806 0x0     - ARP
    >> 0x0800 0x0     - IPv4
    >> 0x86DD 0x0     - IPv6
    >> 0xAAAA 0x0     - CISCO proprietary (STP, PAgP, VTP, PVST+, CDP, DTP and UDLD)
    >> 0x4242 0x0     - CST
```

CONFIG-SET: VACL - Blocks all ICMP echoes & IPv6 on VLAN-162 but forwards all other traffic

```
+-----------------------------------------------------------------------------------------------
|     access-list 101 permit icmp any any echo            - Matches IP ICMP echo
|     !
|     mac access-list extended EtherType
|      permit any any 0x86DD 0x0                          - Matches IPv6 traffic to be denied
|     !
|     vlan access-map VACL 10
|      action drop
|      match ip address 101                               - Drops ICMP Echo
|     vlan access-map VACL 20
|      action drop
|      match mac address EtherType                        - Drops frames matching the ethertype for IPv6
|     vlan access-map VACL 30
|      action forward                                     - Forwards all other traffic
|     !
|     vlan filter VACL vlan-list 162                      - Applies access-map
```

- Switchport Protection

> PVLAN are different from Switchport Protection; PVLANs can span multiple switches whereas protected ports can't.
> Some applications require that no traffic is forwarded between ports on the same switch in the same VLAN.
> The use of protected ports ensures that there are no exchange of unicast, broadcast, or multicast traffic between these ports.
> A protected port does not forward any traffic to any other port that is also a protected port.
> Traffic cannot be forwarded between protected ports at layer2, all traffic passing between protected ports must be forwarded through a layer3 device.
> Forwarding behavior between a protected port and a non-protected port operates as normal.
> If Switchport Protection is configured on an ether-channel, it applies to all ports in the group.
> Configured with "switchport protected".

```
- Switchport Blocking
 > The default behavior of a switch is to forward the packets with unknown destination MAC addresses to all its ports.
 > Switchport Blocking disables this forwarding behavior of unknown uni/multicast addresses on the configured ports.
 > If it's configured on an ether-channel, it applies to all ports in the group.
 > Configured with "switchport block [multicast | unicast]"


-----------
 COMMANDS
-----------
# sh port-security                                      - Shows the counters per secure-port, i.e. MAC, violation count, status
# sh port-security address                              - Shows the current MAC addresses on the port
# sh port-security (interface)                          - Shows more verbose output about the interface specified
# sh dot1x                                              - Shows the dot1x configurations
# sh storm-control                                      - Shows storm-control specifics
# sh ip dhcp snooping                                   - Shows the DHCP snooping configuration
# sh ip source binding                                  - Shows the IP source bindings
# sh ip verify source                                   - Shows the IP source guard configuration


#interface fa0/2                                        >>> Port-Security <<<
 #sw mode {trunk | access}                              - Necessary for switchport security
 #sw port-security                                      - Enables port security (default = 1 MAC allowed)
 #sw port-security {max | vlan | access}                - {max}: Limit the maximum number of MAC address
                                                        - {vlan}: Sets a per-VLAN maximum value
                                                        - {access}: Specifies the VLAN as an access VLAN
 #sw port-security violation {protect|shut|restrict}    - Specifies the violation mode
 #sw port-security aging
 #sw port-security mac-add {mac} [sticky]               - Specifies the secure MAC addresses manually
                                                        - [sticky]: Learn the MAC dynamically but stores it in the running config

                                                        >>> ERR-DISABLE Recovery <<<
#errdisable recovery psecure-violation                  - Enables port recovery for port-security violations
#errdisable recovery {application|all}                  - Enables error disable recovery for application
#errdisable recovery interval {sec}                     - Changes the (def = 300sec) recovery interval
#[no] errdisable detect cause [appl]                    - Enables error disable detection for one or all applications

                                                        >>> 802.1x Authentication <<<
#aaa new-model                                          - Enables AAA
 #aaa authentication login default none                 - Disables AAA for all other authentication methods
 #aaa authentication dot1x [default group radius]       - Creates 802.1x authentication method list querying a radius server
#dot1x system-auth-control                              - Enables 802.1x authentication globally on the switch
#interface fa0/3
 #dot1x port-control auto                               - Enables 802.1x authentication for the port
#ip radius source-interface loopback0                   - (o) Optional source radius traffic from loopback
#radius-server host {ip}                                - (o) Specifies the radius server
#radius-server key {key}                                - (o) Specifies the radius key to use
```

```
                                                     >>> Storm-Control <<<
#storm-control action {shutdown | trap}              - Shuts the interface or sends SNMP trap if a storm occurs

#storm-control {broad | multi | unicast} level [int-threshold] {pps|bps} {value}
                                                     - Enables storm control
                                                     - [int-threshold] is the % of the interface bandwidth (NB:10/100/1000 issue)
                                                     - {pps}: Packets per second
                                                     - {bps}: Bites per second, ONLY available on 3560
                                                     - {value}: pps/bps value

                                                     >>> DHCP Snooping <<<
#ip dhcp snooping                                    - Enables DHCP snooping globally
#[no] ip dhcp relay information option               - Disables (option-82 field) in forwarded DHCP request messages
#interface fa0/3
 #ip dhcp snooping limit rate {pps}                  - Limit untrusted traffic on this interface to {pps}
#interface fa0/4
 #ip dhcp snooping trust                             - Enables a trusted port, needed on ports connected to DHCP server/client

 #ip dhcp snooping vlan {vlan/range}                 - Enables DHCP snooping on a VLAN or range of VLANs

#interface fa0/5                                     >>> IP Source Guard <<<
 #ip verify source [port-security]                   - Enables IP source guard with source IP address filtering
                                                     - [port-security] Enable IP source guard with source IP and
                                                           MAC address filtering

#ip arp inspection vlan {vlan/range}                 >>> DAI (Dynamic ARP Inspection) <<<
#interface fa0/6                                     - DAI is enabled on a per VLAN basis
 #ip arp inspection trust                            - Configures the interface as trusted, (default = untrusted)

                                                     >>> VLAN ACL <<<
#vlan access-map {name} {seq}                        - Creates the access-map for VLAN-ACL
 #match mac address {acl}                            - Matches MAC-address ACL entries or
 #match ip address {acl}                             - Matches IP ACL entries
 #action {drop|forward}                              - Applies a action to the match statement
#vlan filter {name} vlan-list {all | {vlan-id}}      - Applies the VLAN-ACL

#interface fa0/7
 #sw protected                                       - Configures the interface to be a protected port
 #sw block [multicast | unicast]                     - Disables forwarding of unknown uni/multicast addresses out the port
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
   Troubleshooting Switching
*==============================*
- When troubleshooting interfaces and trunks, consider the following:
 > Confirm the state of the interfaces:                                                   # sh int | i line
      >> If an interface is UP/DOWN, is it caused by a speed mismatch?                    # sh int status
      >> Is there a duplex mismatch?                                                      # sh int | i late collisions
 > Is the switchport configured with the correct mode (access/trunk/dynamic)?             # sh int sw | i Name|Admin.*Mode
 > Are both sides of a trunk using the same encapsulation (isl/do1q/negotiated)?          # sh int trunk
      >> Is the correct dot1q native VLAN used?                                           # sh int trunk
      >> Is the dot1q native VLAN the same between two switches on a link?                # sh int trunk
 > Are the pairing of default DTP modes able to negotiate a trunk successfully?           # sh dtp interface | i info|TOT
 > Are the correct interfaces configured to trunk to the correct switches?                # sh int trunk
       > Confirm the switch on the other side of a link.                                  # sh cdp neighbors
 > If a SVI is DOWN/DOWN, does the SVI VLAN exist?                                        # sh vlan brief | i {svi-vlan}
 > If the trunk is connected to a router, was DTP disabled?                               # sh run int {int} | i mode.trunk


- When troubleshooting user VLANs and host issues, consider the following:
 > Are you seeing a host's MAC address on the connected interface?                        # sh mac-add int {int}
 > Are the correct VLAN assigned to an access interfaces (look at 'Vlan')?                # sh int status
 > Are any MAC addresses hardcoded to an interface or null-switched?                      # sh run | i mac.*static
 > Are other switches showing the host's MAC in their CAM table?                          # sh mac-add add {mac}
 > Are any VLANs filtered on trunk links (look at 'Vlans allowed')?                       # sh int trunk
 > Are any ports exceeding the allowed amount of MAC address?                             # sh port-security
 > Are any interfaces in the ERR-DISABLE state?                                           # sh int status
 > Any protected ports preventing communication?                                          # sh run | i interface|protected
 > Any unknown uni/multicast traffic blocked with port-block between switch ports?        # sh run | i interface|block
 > Are any VLAN ACLs configured to drop traffic?                                          # sh run | i vlan-list
 > Is 802.3x flow control disabled?                                                       # sh flowcontrol
 > For more troubleshooting refer to http://routing-bits/2008/11/05/troubleshooting-vlan-issues/


- When troubleshooting VTP, consider the following:
 > Is the same VTP domain name used throughout the VTP domain (name is CaSe-SenSitive)?   # sh vtp status | i Name
 > Are the switches in the correct VTP modes (server/client/transparent)?                 # sh vtp status | i mode
 > Is the MD5 digest the same between switches in a VTP domain?                           # sh vtp status | i MD5
 > Before adding a new switch, confirm its config revision is LOWER than the server's!    # sh vtp status | i Revision
      >> If not change it to zero, by changing mode to transparent and back.              #vtp mode transparent|server


- When troubleshooting dot1q tunnels, consider the following:
 > Was end-to-end layer2 connectivity tested before hand?
 > Was the system MTU increased (1504 bytes) to cater for the metro tag?                  # sh system mtu
 > Was the dot1q tunnel mode specified?                                                   # sh run int {int} | i tunnel.*mode
 > Was the correct metro tag defined?                                                     # sh run int {int} | i access vlan
 > If required was CDP, VTP and STP transport enabled?                                    # sh run int {int} | i l2prot
```

```
- When troubleshooting ether-channels, consider the following:
 > What are the state of the ports and the channel status?                                 # sh ether-channel summary
      >> (U) means the port is in use and (D) means the port is DOWN
      >> (SU) means layer2-channel UP and (SD) means layer2-channel is DOWN
      >> (RU) means layer3-channel UP and (RD) means layer3-channel is DOWN
 > Do both sides use the same channeling protocol?                                         # sh run int {int} | i mode
      >> Are they compatible to negotiate (NOT passive-to-passive, etc.)?
 > Do all member ports have the same configuration?                                        # sh run int {int}
 > Was the configuration done in the correct order? If not delete and do it again!


- When troubleshooting STP, consider the following:
 > Is the expected switch the root bridge for a specific VLAN (Root ID = Bridge ID)?       # sh span vlan {vlan}
      >> If not, which switch is the root bridge (follow the root port!)?                  # sh span vlan {vlan} | i Root
      >> Find the switch attached to that port and repeat until on the root.               # sh cdp nei {root-port}
 > Why was a specific switch elected as root bridge?
      >> Was the default bridge priority changed (default is 32768 + sys-id-ext)?          # sh span vlan 20 | i priority
      >> Was the system ID extension disabled making the switch more preferred?            # sh run | i extend
      >> Remember routers don't use the Sys-id-ext, thus making them root by default!
      >> If none of the above the switch with the highest MAC got elected                  # sh span vlan {vlan} | i Address
 > Not seeing the expected ports in the expected states?                                   # sh span vlan {vlan} | i Root
      >> If not, establish why!                                                            # sh span vlan {vlan} detail
      >> Which port has the lowest cumulative cost to the root (lower = better)?           # sh span vlan {vlan} detail | i cost
            >> A LOCAL root port can be influenced by changing port costs!                 #span vlan {vlan} cost {cost}
      >> Which interface/s goes to the switch with the lowest upstream bridge-ID?          # sh span vlan {vlan} det | i bridg|VLAN
      >> Which port has the lowest port-ID (port priority + port number)?                  # sh span vlan {vlan} det | i desig|VLAN
            >> This can be influenced by the upstream switch's port priority.              #span vlan {vlan} priority {priority}
 > Are any BDPUs filtered, potentially causing STP loops?                                  # sh run | i bpdufilter|backup int
 > Is STP disabled for a specific VLAN?                                                    # sh spanning-tree vlan 20
 > Are any interfaces in the ERR-DISABLE state?                                            # sh int status
 > Is error recovery enabled for the required services?                                    # sh errdisable recovery
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY

```
  ______                                   _____      _
 |  ____|                                 |  __ \    | |
 | |__ _ __ __ _ _ __ ___   ___   ______  | |__) |___| | __ _ _   _
 |  __| '__/ _` | '_ ` _ \ / _ \ |______| |  _  // _ \ |/ _` | | | |
 | |  | | | (_| | | | | | |  __/          | | \ \  __/ | (_| | |_| |
 |_|  |_|  \__,_|_| |_| |_|\___|          |_|  \_\___|_|\__,_|\__, |
                                                               __/ |
                                                              |___/
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-*
|       INDEX             |
*-=-=-=-=-=-=-=-=-=-=-=-=-*
```

```
- Frame-Relay Operation
      + VC (Virtual Circuit)
      + DLCI (DataLink Connection Identifier)
      + LMI (Local Management Interface)
      + LMI Keepalives and N391dte
      + Encapsulation types
      + FECN, BECN and DE
      + PVC Status
      + Broadcast Queue
- Address Resolution
      + Broadcast Replication
      + Static Mappings
      + InARP (Inverse ARP)
            o Disabling Requests
            o Disabling Per DLCI
- Frame-Relay Interfaces
      + Physical Interfaces
      + Point-to-Point Sub-Interface
      + Multipoint Sub-Interface
      + Back-to-Back interfaces
      + MFR (Multilink Frame-Relay)/FRF.16.1
      + Interface States
      + Pinging a Local Frame-Relay Interface
- Partial Mesh NBMA
- Bridging Frame-Relay Links
- Freek (Frame-Relay End-to-End Keepalives)
- Troubleshooting Frame-Relay
      + LMI Communication
      + PVC (Permanent Virtual Circuit) States
      + Back-to-Back Frame-Relay Interfaces
      + Frame-Relay Mappings
      + Partial Frame-Relay Networks
```

Frame-Relay Operation

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS Wide-Area Networking Configuration Guide, Release 12.4T
    > Part 1: Frame-Relay
     > Configuring frame-relay

- Frame-relay is a packet-switching technology commonly implemented as an encapsulation technique, used between LANs over a WAN (Wide Area Network).
- The logical communication paths between two or more DTEs (routers) are called VCs (Virtual Circuits).
- VCs may be permanent (PVCs) or switched (SVCs). PVCs are more common.

- DLCI (DataLink Connection Identifier)
 > DLCIs are used as a frame-relay address to identify the VC over which frames should travel in the frame-relay cloud.
 > A DLCI is contained within a 10-bit field inside the frame-relay header.
 > DLCIs are locally significant to a link and can change as they pass through the network.
 > To see the active DLCIs issue the command "sh frame-relay map".
 > To see all the DLCIs issue the command "sh frame pvc | i DLCI".

- LMI (Local Management Interface)
 > LMI messages manage the communication between the DCE (frame-relay switch) and the DTE (a router).
 > A DTE sends LMI status/inquiry messages to the DCE.
 > The DCE responds with LMI status messages to inform the DTE about the DLCIs and status of each VC.
 > These status/inquiry messages function like and are referred to as LMI keepalives.
 > LMI can be enabled/disabled by using the "keepalive"/"no keepalive" commands.
 > The LMI holdtime is 3x the keepalives and cannot be adjusted directly.
 > However when the keepalive interval is changed the holdtime is changed along with it.
 > If three keepalives (default) are missed an interface will be considered down.
 > There are three LMI types: Cisco/ANSI/q933a.
 > LMI autosense is enabled by default and determines the LMI type to be used.
 > LMI messages/keepalives will inform the router of all of the DLCIs in use, but will not give any information as to which DLCI is associated with which interfaces/sub-interface.
 > LMI is automatically enabled when the command "encapsulation frame-relay" is entered.

- LMI Keepalives and Full Status Update
 > By default, LMI keepalives are sent every 10 sec.
 > Keepalives must match to prevent flapping interfaces.
 > If LMI autosense is unsuccessful, a retry scheme kicks in.
 > Every N391 interval (default is 60 sec, which is six keepalives at 10 sec each), LMI autosense will attempt to ascertain the LMI type and request a complete status update about each VC. This is also known as full status update.
 > If the full status update timers need to be changed then change the N391 interval to how often a full update should be requested.
 > Example: If a router should request a full update once every 180 sec, (180 sec./10 sec. keepalive = 18), only request an update every 18th keepalive.
 > Configured with "frame lmi-n391dte 18" command.

- Routers create frame-relay frames by encapsulating the packet with two additional headers and one trailer.
  - The first header is called the LAPF header, which includes all the fields used by the frame-relay switches to deliver frames across the frame-relay network and this includes the DLCI, DE, BECN and FECN.
  - The second header is called the frame-relay encapsulation header and it contains fields that are only important to the DTE devices. These fields differ between Cisco and IETF encapsulations. The second header also includes a NLPID field (Network Layer Protocol ID) which is commonly used to indicate information about the data-link payload.
  - The frame-relay headers are 8 bytes in size.

- There are two frame-relay encapsulation types: Cisco and IETF.
  - The Cisco option can be used when both DTE devices are Cisco (Cisco encapsulation is used by default).
  - The IETF option is required for multivendor environments.

CONFIG-SET: Examples of Frame-Relay Encapsulations Per-Interface and Per-DLCI

```
+------------------------------------------------------------------------------
|     interface s1/0
|      encapsulation frame-relay ietf                    - Sets IETF encapsulation as default at the interface level
|      frame-relay map ip 10.0.123.2 48 broadcast        - Sets the default configured encapsulation method (IETF)
|      frame-relay map ip 10.0.123.3 49 broadcast cisco  - Per-DLCI encapsulation overwrites per-interface encapsulation
|     !
|     interface s1/1
|      encapsulation frame-relay                         - Sets Cisco encapsulation as default at the interface level
|      frame-relay map ip 10.0.143.2 58 broadcast ietf   - Per-DLCI encapsulation overwrites per-interface encapsulation
|      frame-relay map ip 10.0.143.3 59 broadcast        - Sets the default configured encapsulation method (Cisco)
```

- FECN, BECN and DE
  - FECN (Forward Explicit Congestion Notification) and BECN (Backward Explicit Congestion Notification) are set in the LAPF header to signal congestion on a particular PVC.
  - When a frame-relay switch notices congestion on a PVC, the switch will set the FECN bit indicating congestion in that direction.
  - A router noticing the FECN will set the BECN bit on traffic returning to the source, indicating congestion and notifying the source to slow down its transmission rate.
  - The DE (Discard Eligibility) is used to indicate when traffic ia in violation of the conformed rate and might be subject to discard during periods of congestion. Frames marked with DE bit will be dropped before non-marked frames.
  - Refer to the QOS chapter for more information and configuration about FECN, BECN and DE.

- Frame-Relay PVC Status
  - Active – Both sides of the PVC are up and communicating.
  - Inactive – Local router received status about the DLCIs from the frame-switch. Remote side is down or have config issues.
  - Deleted – Indicates a local config problem. The frame-switch has no such mapping and responds with a 'deleted message'.
  - Static – Indicates that LMI was turned off with the "no keepalives" command.

- Broadcast Queue
  - With large frame-relay networks huge amounts of DLCI updates can consume bandwidth, interface buffers and even cause packet loss.
  - To avoid such problems, a special broadcast queue can be created on an interface to use its own queue and buffers.

- CDP is enabled by default on all supported interfaces (except for frame-relay multipoint sub-interfaces).

```
-----------
 COMMANDS
-----------
# sh frame-relay map                                   - Shows the DLCI mappings, status, dynamic/static, type, broadcast, etc
# sh frame-relay pvc [dlci]                            - Shows PVC status, DLCIs, in/output packets, PVC uptime, etc

# debug frame-relay packet                             - Shows the DLCI mappings
                                                       - Should actually be 'debug frame-relay frame', not packet :)
                                                       - 'encaps failed- no map entry" shows incorrect DLCI assignment

#interface s0/1
 #encapsulation frame-relay [ietf]                     - Enables frame-relay encapsulation on the physical interface
                                                       - [ietf] Use RFC1490/RFC2427 encapsulation (default = Cisco)
 #frame-relay lmi-type {cisco|ansi|q933a}              - Changes the LMI type (default = Cisco)
 #keepalive {number}                                   - Sets the LMI keepalive interval (default = 10 sec.)
 #frame lmi-n391dte {number}                           - Sets a full status update polling interval
 #frame broadcast-queue {q-size} {bps} {packet-rate}   - Creates a broadcast queue for the interface
 #cdp enable                                           - Enables CDP on the interface


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
  Address Resolution
*======================*
- Frame-relay networks are multi-access networks, which means that more than two devices can attach to the network, similary to LANs.
- Unlike LANs, you cannot send a data link layer broadcast over frame-relay. Therefore frame-relay networks are often called
      NBMA (Non-Broadcast Multi-Access) networks.
- Because frame-relay is a multi-access technology, it always requires layer3-to-layer2 address resolution to identify to which
      remote router a frame is destined for.
- The exceptions are frame-relay point-to-point sub-interfaces and PPP-over-frame-relay.

- Broadcast Replication
 > Frame-relay does not have the capability to send a single frame over multiple PVCs to multiple destinations.
 > But the broadcast functionality is still sometimes required by routing protocols.
 > Also known as a pseudo-broadcast, frame-relay can make duplicate copies of a packet and send one on each PVC.
 > Frame-relay can thus send copies of layer3 broadcasts over VCs, if configured to do so.

- Static Mappings
 > Are used to statically resolve a REMOTE layer3 address(IP) to a LOCAL layer2 address(DLCI).
 > Are manually configured with the command "frame-relay map".
 > Require broadcast to be enabled manually, if needed.
 > Static frame-relay mappings (frame-relay map) override dynamic mappings (learned via InARP).

- InARP (Inverse ARP)
 > Is used to dynamically resolve a REMOTE layer3 address(IP) to a LOCAL layer2 address(DLCI).
 > Is enabled automatically when an IP address is configured.
 > Has auto-broadcast enabled by default.
 > The InARP status query request can be disabled per DLCI or for all DLCIs on an interface. The InARP reply cannot be disabled!!
 > The command "no frame-relay inverse-arp" configured on a physical interface stops the InARP query messages only for the
      physical interface, not the sub-interfaces. It must be configured on the sub-interfaces if needed.
```

> When a point-to-point interface is connected to an InARP disabled interface, the InARP disabled interface will still reply, provided an IP address is configured on that interface. On the querying router the "sh frame-relay map" will still show that mapping as dynamic.

- To force/trigger an interface to InARP:
  > The interface can be "shutdown", "no shutdown" or,
  > The InARP mappings can be manually cleared with "clear frame inarp".

```
-----------
 COMMANDS
-----------
# sh frame-relay map                             - Shows the DLCI mapping, status, dynamic/static, type, broadcast

# clear frame-relay inarp                        - Clears the dynamic InARP mappings and forces InARP

#interface s1/0
 #encapsulation frame-relay
 #no frame-relay inverse-arp                     - Disables InARP requests for the interface
 #no frame-relay inverse-arp ip {dlci}           - Disables InARP requests only for the DLCIs specified
 #frame-relay map ip {ip} {dlci} [broadcast]     - Statically maps a remote IP address to a local DLCI
                                                 - [broadcast] Enables frame-relay broadcast relay across the PVC
```

-=-

==============================

Frame-Relay Interfaces

==============================

- Frame-relay interfaces carry one of two characteristics: point-to-point or multipoint.

- Physical Interfaces
  > Are treated as multipoint interfaces.
  > Multipoint means the interface can terminate multiple PVCs (layer2 circuits).
  > Requires layer3-to-layer2 resolution through either InARP or manual mapping.
  > Manual mapping per PVC is done with the "frame map ip" command.
  > To manually assign just one PVC on the interface use "frame-relay interface-dlci".

- Point-to-Point Sub-Interfaces
  > Can only terminate one PVC.
  > Do not require layer3-to-layer2 resolution, since there is only one PVC.
  > Do not send InARP status queries, but will respond to an InARP status query request.

- Multipoint Sub-Interfaces
  > Are treated as multipoint interfaces.
  > Can terminate multiple PVCs.
  > Requires layer3-to-layer2 resolution through either InARP or manual mappings.
  > Manual mapping per PVC is done with the "frame map ip" command.
  > To manually assign just one PVC on the interface use "frame-relay interface-dlci".

```
- Back-to-Back (NNI) Frame-Relay Interfaces
 > Are router-to-router serial links running frame-relay encapsulation, but with no frame-relay switch inbetween to do LMI.
 > For back-to-back links two things are required:
      >> Disable LMI keepalives with "no keepalives".
      >> Configure one side as a DCE end with a clock rate.
 > Any DLCIs can be used, provided both sides have the same DLCIs configured.
```

```
CONFIG-SET: Frame-Relay Interface Types
+-----------------------------------------
|     interface s1/0                                     >>> Physical Interface <<<
|      encapsulation frame-relay ietf                    - Enables IETF encapsulation
|      ip address 10.0.3.1 255.255.255.0                 - Configuring an IP enables InARP automatically
|      frame-relay map ip 10.0.3.2 103                   - Configures a static DLCI mapping (use DLCI-103 to reach 10.0.3.2)
|      frame-relay map ip 10.0.3.5 105 broadcast         - Enables broadcast notification for the host
|     !
|     interface s1/1
|      encapsulation frame-relay                         - Frame-relay encapsulation must be enabled on the physical
|     !
|     interface s1/1.1 point-to-point                    >>> Point-to-Point Sub-Interface <<<
|      ip address 10.0.1.4 255.255.255.0
|      frame-relay interface-dlci 104                    - Assigns DLCI-104 to the interface  (only one PVC)
|     !
|     interface s1/1.2 multipoint                        >>> Multipoint Sub-Interface <<<
|      ip address 10.0.2.4 255.255.255.0                 - This interface will rely on the dynamic InARP mappings received
|     !
|     interface s1/2                                     >>> Back-to-Back interface <<<
|      ip address 10.1.5.1 255.255.255.0
|      encapsulation frame-relay                         - Cisco encapsulation is enabled by default
|      no keepalives                                     - Disables LMI keepalives
|      clock rate 256000                                 - Sets this interface as the point-to-point DCE
```

```
- MFR (Multilink Frame-Relay) or FRF.16.1
      > DOC-CD LOCATION
       > 12.4T Configuration Guide > WAN
        > Cisco IOS Wide-Area Networking Configuration Guide, Release 12.4T
         > Part 1: Frame-Relay
          > Multilink Frame-Relay (FRF.16.1)

 > MFR provides a cost-effective way to increase bandwidth by enabling multiple frame-relay links to be aggregated into a
      single bundle of bandwidth acting as one interface.
 > MFR variable bandwidth support allows the option to activate or deactivate a frame-relay bundle based on Class-A, -B, or -C.
 > Class A (Single Link)
      >> The bundle will activate when any single bundle link is up and will deactivate when all bundle links are DOWN (default).
 > Class B (All Links)
      >> The bundle will activate when all bundle links are up and will deactivate when any single bundle link is DOWN.
 > Class C (Threshold)
      >> The bundle will activate when the minimum configured number of bundle links are up and will deactivate when the
             minimum number of configured bundle links fails to meet the threshold.
```

CONFIG-SET: MFR - Multilink Frame-Relay (FRF.16.1)

```
+-----------------------------------------------------
|     interface mfr1.1 point-to-point                      - Creates the multilink frame-relay interface
|      ip address 10.5.12.9 255.255.255.0                  - Assigns the logical interface an IP address
|      frame-relay interface-dlci 102                      - Assigns the PVC identifier
|      multilink bandwidth-class b                         - Both links must be up before the bundle is brought up
|     !
|     interface s0/2/1
|      no ip address
|      encapsulation frame-relay mfr1                      - Assigns the first interface to the bundle
|     !
|     interface s0/2/0
|      no ip address
|      encapsulation frame-relay mfr1                      - Assigns the second interface to the bundle
|
```

- Interface States
  > The physical interface connected to a frame-relay switch will be UP/UP, once it receives LMI from that frame-relay switch, regardless of the DLCI it is learning or not learning.
  > This means a physical interface can be UP/UP, even though there is no layer2 communication.
  > But with a point-to-point sub-interface, the sub-interface will only show UP/UP, when LMI is received AND one of the received DLCIs matches the DLCI configured on the sub-interface.
  > When a multipoint sub-interface has multiple DLCIs defined, all DLCIs must be down before the interface will show DOWN/DOWN. If one DLCI is up, the interface will be UP/UP.
  > http://routing-bits/2009/01/26/frame-relay-interface-states/

- When removing a frame-relay sub-interface configuration, the configuration is removed off the interface, but the sub-interface will only be deleted after a reboot.
- This can be seen with a "sh ip int brief" when the interface is listed as DELETED.
- Thus to change a sub-interface from point-to-point to multipoint, delete the sub-interface and reload the router. Then create new multipoint interface.

!!NOTE!! Always do "show frame-relay map" when starting a lab and after configuration is complete to verify layer2 connectivity. If there are 0.0.0.0 frame-relay mappings, save the configuration and reload. It is the only way to get rid of this.

- To ping a locally configured IP on a frame-relay interface, layer3-to-layer2 resolution is required. This is needed because the frame actually exits the router to the other side of the link only to get redirected back because of the remote IP. If the mapping is not done, the ping reply is dropped by the router on the other side of the link.

CONFIG-SET: Pinging the local IP on frame-relay interface

```
+----------------------------------------------------------
|     interface s0/1
|      ip address 10.5.34.3 255.255.255.0                  - Configures the interface IP
|      encapsulation frame-relay
|      frame-relay map ip 10.5.34.4 304 broadcast          - Maps the remote-end IP to local-DLCI
|      frame-relay map ip 10.5.34.3 304                    - Maps the local IP to local-DLCI
|
-----------
 COMMANDS
-----------
# sh frame-relay map                                   - Shows the DLCI mappings, status, dynamic/static, LMI types
# sh frame-relay multilink                             - Shows the current frame-relay multilink configuration
# sh interfaces mfr {mfr-interface}                    - Shows information and packet statistics for the bundled interfaces

#interface s0/1
 #encapsulation frame-relay                            - Enables frame-relay
#interface s0/1.345 {point-to-point|multipoint}        - Sets the type of sub-interface
 #frame-relay interface dlci {dlci}                    - Used when only one PVC terminates on the interface
 #frame-relay map {prot}{ip}{dlci}[broadcast]          - Used when one or more PVCs terminate on the interface
                                                       - Statically maps a remote IP address to a local DLCI
                                                       - [Broadcast] is not enabled by default with the "map" command


#interface s0/2
 #no keepalive                                         - Disables the LMI keepalive interval on a back-to-back interface
 #clock rate {bps}                                     - Enables the DCE end to provide clocking


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*================================================*
   Partial Mesh NBMA (Non-Broadcast Multi-Access)
*================================================*
- Frame-relay sub-interfaces provide a mechanism for supporting partially meshed frame-relay networks.
- Spokes cannot resolve each other via InARP, because the endpoints don't have layer2 circuits provisioned between them.
- Hub-and-Spoke is a type of partial mesh NBMA network.

- Scenario
 > R2 connects to the frame-relay cloud using one interface and two PVCs. One to R1 and one to R3.
 > Thus R2 is the 'hub' and R1/R3 are 'spokes'.
 > Although not direct linked, R1 and R3 still require layer2 reachability.
 > There are four possible solutions:
      >> Add additional static mappings via the hub router for the other spokes.
      >> Change to point-to-point sub-interfaces.
      >> Use static IP routing with next-hop instead of interface.
      >> Use layer3 dynamic routing, such as OSPF interface type point-to-multipoint.

CONFIG-SET: Frame-Relay Hub-and-Spoke Example with Static Mappings
+-------------------------------------------------------------------
|R2 is configured as the hub. R1 and R3 as spokes.
|
|R2#
|    interface s2/0                                          - R2 is the hub
|     encapsulation frame-relay
|     ip add 10.5.0.2 255.255.255.0
|     frame-relay map ip 10.5.0.1 201 broadcast              - Static mapping to each spoke allowing broadcast replication
|     frame-relay map ip 10.5.0.3 203 broadcast              - Static mapping to each spoke allowing broadcast replication
|
|R1#
|     interface s1/2                                        - R1 is a spoke
|      encapsulation frame-relay
|      ip add 10.5.0.1 255.255.255.0
|      frame-relay map ip 10.5.0.2 102 broadcast            - Static mapping to the hub
|      frame-relay map ip 10.5.0.3 102                      - Static mapping to the other spoke via the hub
|                                                           - Broadcasts wont be carried beyond R2, so no need to enable it
|R3#
|     interface s0/2                                        - R3 is a spoke
|      encapsulation frame-relay
|      ip add 10.5.0.3 255.255.255.0
|      frame-relay map ip 10.5.0.2 302 broadcast            - Static mapping to the hub
|      frame-relay map ip 10.5.0.1 302                      - Static mapping to the other spoke via the hub
|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===================================*
   Bridging Frame-Relay Links
*===================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Wide-Area Networking Configuration Guide, Release 12.4T
         > Part 1: Frame-Relay
          > Customizing Frame Relay for Your Network
           > Configuring Frame Relay Subinterfaces
```

- The frame-relay bridging uses the same spanning-tree algorithm as the other bridging functions.
- The bridging spanning tree views each PVC as a separate bridge port.
- A frame arriving on one PVC will be relayed back out on a separate PVC on the same physical interface.

```
CONFIG-SET: Bridging Frame-Relay Sub-Interfaces
+-----------------------------------------------------
| This shows frame-relay DLCIs 42 and 64 as separate point-to-point links with transparent bridging.
| The local router will thus be transparent to the routers connected to each PVC.
|
|     bridge irb                                            - Creates transparent bridge group
|     bridge 1 protocol ieee
|     !
|     interface s0
|      encapsulation frame-relay                            - Enables frame-relay transparent bridging
|     !
|     interface s0.1
|      bridge-group 1                                       - Associates the sub-interface with a bridge group 1
|      frame-relay map bridge 42 broadcast                  - Bridges DLCI 42 and 64 together
|     !
|     interface s0.2
|      bridge-group 1                                       - Associates the sub-interface with a bridge group 1
|      frame-relay map bridge 64 broadcast                  - Bridges DLCI 42 and 64 together
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================================*
   Freek (Frame-Relay End-to-End Keepalives)
*==============================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Wide-Area Networking Configuration Guide, Release 12.4T
        > Part 1: Frame-Relay
         > Configuring Frame-Relay
          > Customizing Frame Relay for Your Network
           > Configuring Frame-Relay End-to-End Keepalives

- Freek provides the ability to track end-to-end reachability between DTE devices.
- Freek can be configured on a physical interface, but when the freek status goes down, the router will not bring the physical
      interface DOWN, because the router will not know when to bring it back up.
- For this reason it is recommended to configure freek on a sub-interface.
- Copy descretly owned by Kane  Bagwell
- Freek Modes:
 > Bidirectional
      >> Both sides of the PVC can send and respond to keepalive requests.
      >> If one side is configured as Bidirectional, the other end must be configured in the same way.
      >> Sets the timers and keeps track of error counters.
 > Request
      >> With Request mode only one side is enabled in Send mode.
      >> If one side is configured as Request, the other end must be Reply or Passive-Reply.
      >> Sets the timers and keeps track of error counters.
 > Reply
      >> The device waits for and replies to keepalive requests.
      >> If one side is configured as Reply, the other end must be Request.
      >> Sets the timers and keeps track of error counters.
 > Passive-Reply
      >> The device waits for keepalive requests and responds to them.
      >> Sets the timers.

 ----------
 COMMANDS
 ----------
# sh frame-relay pvc                                   - Shows the freek status as 'EEK UP' or 'EEK DOWN'
# sh frame-relay end-to-end keepalive                  - Shows frame-relay end-to-end VC keepalive information

#map-class frame-relay FREEK                           - Creates a map-class
  #frame-relay end-to-end keepalive mode {bidirectional | request | reply | passive-reply}
                                                       - Enables freek for the class
  #frame-relay end-to-end keepalive timer recv {sec}   - Sets interval timer for incoming end-to-end keepalive requests
  #frame-relay end-to-end keepalive timer send {sec}   - Sets interval timer for outgoing end-to-end keepalive requests

#interface s1/0.345 {point-to-point|multipoint}
 #frame-relay class FREEK                              - Applies the map-class to EACH DLCI on the interface, OR
 #frame-relay interface-dlci 402
  #class FREEK                                         - Applies the map-class ONLY to DLCI-402
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===================================*
   Troubleshooting Frame-Relay
*===================================*
- When troubleshooting LMI communication, consider the following:
 > Is the physical interface connected and unshut (should be at least UP/DOWN)?                # sh ip int brief
 > To see all the DLCIs received issue the command                                             # sh frame pvc | i DLCI
 > Does the frame-relay encapsulation match between neighbors (Cisco or IETF)?                  # sh run | i encap.*frame
 > Is there two way LMI communication (both 'Sent' and 'Rcvd' should be non-zero)?             # sh frame lmi int {int} | i Sent
 > Does the LMI type match between neighbors (if type mismatch, 'yourseen' will be 0)?         # debug frame lmi
 > Was LMI disabled with "no keepalive" on a non back-to-back interface?                       # sh run | i interface|no keepalive
      >> This could cause a link to show UP/UP even though it's not.                           # sh frame pvc | i STATIC
 > If a physical interface is connecting to the frame-relay switch,
      >> the interface will be UP/UP once it receives LMI, even if there are no valid DLCIs.    # sh frame pvc int {int}
 > If a point-to-point sub-interface is connecting to the frame-relay switch,
      >> the interface will only show UP/UP when it receives LMI with a matching DLCI.          # sh frame pvc int {int}
 > If a multipoint sub-interface is connecting to the frame switch,
      >> all DLCIs must be DOWN before the interface will show DOWN/DOWN.                       # sh frame pvc int {int} | DLCI

- PVC (Permanent Virtual Circuit) States:                                                       # sh frame pvc | i DLCI
 > ACTIVE   - Both sides of the PVC are up and communicating.
 > INACTIVE - Local router received LMI status from frame-switch. Remote router is down or have config issues.
 > DELETED  - Local config problem. The frame-switch has no such mapping and responds with 'deleted' status.
 > STATIC   - LMI keepalives were disabled with "no keepalive".

- For back-to-back frame-relay interfaces, consider the following:
 > Firstly confirm which end is the DCE and which side is the DTE.                             # sh controllers {int} | i DCE|DTE
 > Secondly confirm the DCE end is providing clocking.                                          # sh run | i interface|clock rate
 > Have keepalives been disabled (Required for back-to-back)?                                   # sh run | i interface|no keepalive
 > Are both sides using the same DLCIs (Required for back-to-back)?                             # sh frame pvc | i DLCI

- When troubleshooting frame-relay mappings, consider the following:
 > For successful mappings, the PVCs should be in ACTIVE state.                                 # sh frame pvc | i DLCI
 > To see active DLCIs and their mappings issue, use the command:                               # sh frame map
 > If sub-interfaces were removed to be re-used, was a reload done after deletion?              # sh ip int brief | i deleted
 > If there are 0.0.0.0 frame-relay mappings, then save the config and reload.                  # sh frame map
 > For point-to-point sub-interfaces, was the interface DLCI correctly specified?               # sh run | i interface.*dlci
 > For multipoint interfaces
      >> Is inverse-ARP relied on to do the mappings?                                           # sh frame map | i dynamic
      >> If not, was the mapping done statically?                                               # sh frame map | i static
            >>> Are the static mappings defined correctly?                                      #frame map ip {peer-ip} {local-dlci}
            >>> Where needed was broadcast replication enabled on the static mappings?          # sh run | i frame.*broadcast
 > 'Encaps failed--no map entry link' indicates a mapping error.                                #debug frame packet

 > A typical issue with partial frame-relay networks is mapping issues:
      >> Inverse-ARP can only be used between directly connected frame neighbors!
      >> Indirect neighbors should use either static mapping or point-to-point sub-interface!
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY

```
 ______    ______    ______
|  ___ \  |  ___ \  |  ___ \
| |___) ) | |___) ) | |___) )
|  ____/  |  ____/  |  ____/
| |       | |       | |
|_|       |_|       |_|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-*
|      INDEX            |
*-=-=-=-=-=-=-=-=-=-=-=-*
```

- PPP Overview
- PPP Operation
  - + LCP (Link Control Protocol)
  - + NCP (Network Control Protocol)
  - + IPCP (Internet Protocol Control Protocol)
  - + PPP Phases
  - + IPCP Default Route
- Peer Address Allocations
  - + Peer Default Address
  - + Local Address Pool
  - + DHCP
  - + TACACS+
- Peer Neighbor Route
- PPP Authentication
  - + Putting a "?" in Password
  - + PAP (Password Authentication Protocol)
  - + CHAP (Challenge Handshake Authentication Protocol)
  - + EAP (Extensible Authentication Protocol)
- MPPE (Microsoft Point-To-Point Encryption)
- PPP Reliable Link
- LQM (Link Quality Monitoring)
- MLP (Multilink PPP)
  - + LFI (Link Fragmentation and Interleaving)
  - + MCMP (Multiclass Multilink PPP)
  - + MRRU Negotiation
- PPPoFR (PPP over Frame-Relay)
- PPPoE (PPP over Ethernet)
  - + Using a Local IP Pool
  - + Using DHCP
- PPP Half-Bridging
- Troubleshooting PPP
- OUTPUT-101
  - + PPP Negotiation Explained using a Debug
  - + One-Way PAP Authentication Explained using a Debug
  - + One-Way CHAP Authentication Explained using a Debug
  - + Other Debug Outputs Explained

PPP Overview

- PPP (Point-to-Point Protocol) is a suite of protocols operating at the data link layer, which are used in establishing a connection between two networking nodes over a variety of different physical layer connections.
- PPP was designed to carry traffic over synchronous and asynchronous links.
- PPP acts as the interface between the internet protocol layer and the physical link layer.

- PPP is a popular layer2 WAN technology, due to its rich feature set that includes the following:
  > A comprehensive framing mechanism with built-in error detection.
  > Monitoring the quality of a link prior to the sending of a frame.
  > Capability to encapsulate traffic over other layer2 WAN technologies such as ethernet, frame-relay and ATM.
  > Offers authentication using various authentication protocols including PAP, CHAP and EAP.
  > Extendable to use additional optional features, including compression, encryption and link aggregation.

PPP Operation

- The PPP standard (RFC 1661 and others) describes three main components:
  > Encapsulation Method
    >> PPP takes higher-layer datagrams such as IP and encapsulates them for transmission over the underlying physical layer link.
    >> PPP defines a specific frame format for encapsulating data that is based on the HDLC framing method.
    >> A PPP frame is small in size and contains only simple fields to maximize bandwidth efficiency and speed.
  > LCP (Link Control Protocol)
    >> LCP is responsible for setting up, maintaining and terminating the link between routers.
    >> LCP is a flexible, extensible protocol that exchanges configuration parameters to ensure that both end-routers agree on how the link will be used.
    >> LCP may invoke an authentication protocol, if so configured.
  > NCP (Network Control Protocol)
    >> After the general link setup is completed with LCP, control is passed to the NCP-specific layer3 protocol being carried across the PPP link.
    >> When IP is carried over PPP, the NCP used is the IPCP (Internet Protocol Control Protocol).
    >> IPCP performs the required network-layer-specific configurations before the link can carry any IP traffic.
    >> The remainder of this chapter will focus only on IPCP, and no other NCPs.

- IPCP (Internet Protocol Control Protocol)
  > IPCP performs similar functions to those of LCP (IPCP link setup, maintenance and termination).
  > Where LCP is responsible for the underlying link, IPCP is ONLY responsible for the IP link (portion) of the connection.
  > IPCP uses the same packet formats (described below) as LCP.
  > Think of IPCP as a 'lite' version of LCP.
  > An IPCP link runs over an LCP link.

```
- Some relevant LCP packet formats:
 > Configure-Request
      >> Is sent by the router wishing to open a connection.
 > Configure-Ack
      >> Indicates acknowledgment if every configuration option received in a Configure-Request was recognized and agreed on.
 > Configure-Nak
      >> Indicates that some of the configuration options received in a Configure-Request were not agreed on (not acknowledged).
 > Configure-Reject
      >> Indicates that some of the configuration options received in a Configure-Request were not recognized.
 > Terminate-Request
      >> Is used by the router wishing to close a connection.
 > Terminate-Ack
      >> Is sent in response to a Terminate-Request to close a connection.


- The PPP finite state machine (process of setting up, using and closing a PPP link) can be described as follow:
 > Link Dead Phase
      >> A PPP link always begins and ends in this phase.
      >> In this phase there is no physical layer link established between the two routers.

 > Link Establishment Phase
      >> In this phase the physical layer is connected and LCP attempts a basic link setup.
      >> Router-A sends an LCP Configure-Request message to router-B specifying the parameters it wishes to use.
      >> Any of the following options could be included to be agreed upon:
            >>> MRU (Maximum-Receive-Unit) is the maximum datagram size.
            >>> Authentication-protocol to use, if any.
            >>> Quality-protocol to enable quality monitoring of the link.
            >>> Magic-Number is used to detect looped links or other anomalies.
            >>> Multilink PPP which adds several of its own options (covered in a later section).
      >> If router-B agrees to all of the requested options, it will reply with a Configure-Ack.
      >> If router-B doesn't agree, it will reply back with a Configure-Nak.
      >> If router-B doesn't recognize some of the options, it will reply with Configure-Reject.
      >> If router-A and B agree on the parameters, the LCP is considered open and the phase initiated.
      >> If router-A and B cannot agree on any parameters, the physical link is terminated and the phase reset to
         the Link Dead phase.

 > Optional Authentication Phase
      >> If authentication is configured, the configured protocol will be employed.
      >> If configured and authentication is successful, the link proceeds to the IPCP phase.
      >> If configured and authentication is not successful, the link fails and transitions to the Link Termination phase.

 > Network-Layer Protocol Phase
      >> Once the basic link setup is completed, IPCP is used to set up an IP NCP link between the two routers.
      >> This is done using the IPCP Configure-Request messages to configure the following options:
            >>> IP-Address                      - Used to request an IP address or to send the used IP address.
            >>> IP-Compression-Protocol         - Allows routers to negotiate the use of TCP and IP header compression.
      >> The receiving router can send back an IPCP Configure-Ack, an IPCP Configure-Nak, or an IPCP Configure-Reject.
      >> If CDP is enabled, CDP negotiation also occurs in NCP phase.
      >> After the IPCP phase is complete, the link proceeds to the Link Open state.
```

> Link Open Phase
>> In this state the LCP link and IPCP links are open and operational, either router may send packets as required.

> Link Termination Phase
>> Is based either on link failure or by either end-router wanting to terminate communication.
>> The router wanting to terminate the link sends a LCP termination frame and the receiving router acknowledges it.
>> The link then goes to the Link Dead phase.

- IPCP Default Route
> PPP can also insert a dynamic default route whenever IPCP negotiations succeed (and remove it when the line protocol goes down).
> This can only be configured using a PPP virtual template interface.
> A static route cannot be configured through the virtual-template interface.
> The client router may use this default route to access external destinations without requiring any local routing.
> Configured with "ppp ipcp route default" under a virtual-template interface.

```
-----------
 COMMANDS
-----------
#interface s1/1
 #ppp lcp predictive                                     - Reduces negotiation time by predicting responses from peers
 #ppp ipcp predictive                                    - Reduces negotiation time by predicting responses from peers
 #ppp lcp fast-start                                     - Interface responds immediately once a connection is established

#interface virtual-template1
 #ip address negotiated                                  - Specifies that the IP address is negotiated
 #ppp ipcp default route                                 - Configures a default route through a PPP virtual access interface
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
   Peer Address Allocations
*============================*
```

- DOC-CD LOCATION
> 12.4T Configuration Guides
> Cisco IOS Dial Technologies Configuration Guide, Release 12.4T
> Part 9: PPP Configuration
> Configuring Media-Independent PPP and Multilink PPP
> IP Address Pooling

- An IP address of a PPP interface can be manually configured or negotiated during IPCP negotiation.
- If negotiated, the IP address may be provided to the remote router across a point-to-point link using several methods:
> Peer Default Address
>> Router-A could be configured to present a peer IP address to router-B.
>> If router-B has no address assigned, the presented address is acknowledged and used.
>> If router-B already had an address assigned, the assigned address is used since router-B won't request an IP address.
>> Configured with "peer default ip address".

```
 > Local Address Pool
      >> Router-A could be configured to serve an IP address from a locally configured pool (up to 256 addresses).
      >> If router-B requests an IP address, router-A will assign the first available unassigned IP address from the pool.
      >> Configured with "ip local pool" and "peer default ip address pool".
      >> Example covered in the PPPoE Section's Config-Set.

 > DHCP
      >> Router-A could be configured as a host-based DHCP server to accept and process DHCP requests
            from DHCP clients like router-B.
      >> Configured with "ip dhcp pool" and "peer default ip address dhcp-pool".
      >> Example covered in the PPPoE Section's Config-Set.

 > TACACS+
      >> During the authorization phase of IPCP address negotiation, TACACS+ could return an IP address for
             the authenticated interface.
      >> A TACACS-provided IP address will override a default peer IP address.
      >> The TACACS implementation is beyond the scope of the CCIE, but be aware of this for real world networks.
```

```
CONFIG-SET: PPP Peer Default Address Allocation
+--------------------------------------------------
|R1#
|     interface s0/2
|      ip address 10.5.0.1 255.255.255.0
|      encapsulation ppp                                   - Enables PPP encapsulation on the interface
|      peer default ip address 10.5.0.2                    - Specifies the IP address to be issued to R2
|
|R2#
|     interface s0/1
|      ip address negotiated                               - R2 will request an IP address from the other side of the link
|      encapsulation ppp
|      clock rate 2000000                                  - Required on the DCE end of the link
|
|
>     R2#sh ip int brief | i Serial0/2
>     Serial0/2     10.5.0.2   YES  IPCP   up  up          - Notice the method is IPCP
>
```

```
----------
 COMMANDS
----------
#ip local pool {name} {ip} [end-ip]                        - Creates a local IP pool of one or more IP addresses

#ip dhcp excluded-address {ip}                             - Always excluded already used IP addresses
#ip dhcp pool {name}                                       - Creates a local DHCP-pool of one or more IP addresses

#interface s1/0
 #peer default ip address {ip}                             - Offers the configured IP address to the peer
 #peer default ip address pool {name}                      - Offers an IP address from the local IP pool
 #peer default ip address dhcp-pool {name}                 - Offers an IP address from the local DHCP pool
```

Peer Neighbor Route

```
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Dial Technologies Configuration Guide, Release 12.4T
         > Part 9: PPP Configuration
          > Configuring Media-Independent PPP and Multilink PPP
           > Disabling or Reenabling Peer Neighbor Routes
```

- Another part of the dynamic allocation of IPCP is a feature known as Peer Neighbor Route.
- It's a mechanism to insert a dynamic host route (/32) for the peer's IP address once that peer's IP address is learned.
- It is useful to provide reachability when both ends of the same PPP link are not in the same logical subnet, e.g. IP-unnumbered.

- Consider the following scenario:
 > R1 S1/0 is directly connected to R2 S2/0.
 > R1 S1/0 uses an IP address of 1.1.1.1/8.
 > R2 S2/0 uses an IP address of 2.2.2.2/16.
 > R2 will have 1.1.1.1 in the routing table as 1.1.1.1/32.
 > R1 will have 2.2.2.2 in the routing table as 2.2.2.2/32.
 > Since only the IP address (no mask) was advertised, a /32 is assumed for reachability on both routers.
 > R1 will be able to ping 2.2.2.2 as a result.

- PPP peer neighbor route can be safely disabled when both ends of the link are in the same logical subnet.
- If IP-unnumbered or dissimilar IP subnets are used on a point-to-point PPP link, leave it enabled.
- The IP routing table must be cleared or the interface must be flapped to remove the PPP-generated host route.

COMMANDS

```
#interface s2/0
 #no peer neighbor-route                                - Disables the dynamic /32 routes for a neighbor's IP addresses
```

PPP Authentication

```
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Dial Technologies Configuration Guide, Release 12.4T
         > Part 9: PPP Configuration
          > Configuring Media-Independent PPP and Multilink PPP
           > Enabling CHAP or PAP Authentication
```

- RFC 1334 defines the original PPP authentication protocols: PAP and CHAP.
- RFC 2284 defines the third PPP authentication protocol: EAP.

```
- The authentication protocol is negotiated during the Link Establishment Phase in Configure-Request packets.
- If a ? (question-mark) is part of a password, use CRTL-V or ESC-Q to enter it on the CLI.

- RFC Terminology
 > Authenticator
      >> Is the end of the link requiring the authentication.
      >> The authenticator specifies the authentication protocol to be used in the Configure-Request during the Link
            Establishment phase.
 > Peer
      >> Is the other end of the point-to-point link.
      >> A username/password pair is sent by the peer to the authenticator.

*-----------------------------------------*
  PAP (Password Authentication Protocol)
*-----------------------------------------*
 > PAP authentication uses a two-way handshake to establish a connection.
 > PAP sends clear text usernames and clear text passwords.

 > There are three different types of PAP packets:
      >> Authenticate-Request (Auth-Req)
            >>> The peer transmits the Authenticate-Request packet during the Authentication Phase.
            >>> Auth-Req is used to start the PAP authentication and is transmitted/accepted only during the Authentication Phase.
            >>> The Authenticate-Request packets are repeatedly sent until a valid reply packet is received.
      >> Authenticate-ACK (Auth-Ack)
            >>> If the username/password pair received in an Authenticate-Request packet is acceptable, the authenticator replies
                  with an Authenticate-ACK packet.
      >> Authenticate-NAK (Auth-Nak)
            >>> If the username/password pair received are not acceptable, the authenticator replies with an Authenticate-NAK packet.

 > PAP supports unidirectional (one-way) and bi-directional (two-way) authentication.
 > With unidirectional authentication, only the side (authenticator) receiving the Auth-Req authenticates the remote side (peer).
 > The peer does not authenticate the authenticator.

 > With bi-directional authentication, each side independently sends an Auth-Req packet.
 > Both sides respond with either an Auth-Ack or Auth-Nak.
 > It could happen that only one directions authentication was successful and the reverse direction not.

 > The authenticator requires a globally configured username and password.
 > The username/password pair should match those sent by the peer.
 > With PAP, the "username {uid} password {pwd}" is used only to verify that an incoming username and password are valid.
```

```
CONFIG-SET: PPP one-way PAP authentication
+--------------------------------------------------------
|Example: R2 connects to R1, where R1 authenticates R2
|Refer to Output-101 section for debug output from this configuration
|
|R1#  username R2-UID password ccie                    - Received Auth-Req must match local usernames/passwords
|     !
|     interface s1/1
|      ip address 10.5.1.1 255.255.255.0
|      encapsulation ppp
|      ppp authentication pap                             - Enables R1 as the authenticator
|     !
|
|R2#  interface s1/1
|      ip address 10.5.1.2 255.255.255.0
|      clock rate 64000
|      encapsulation ppp
|      ppp pap sent-username R2-UID password ccie         - R2 (peer) will send this in its Auth-Req
|
>
>R1#sh users                                              - Displays the authenticated session
>       Interface    User      Mode        Idle       Peer Address
>       Se1/1        R2-UID    Sync PPP    00:00:02   10.5.1.2
>
```

```
CONFIG-SET: PPP two-way PAP authentication
+-------------------------------------------------------
|Example: R2 connects to R1, where R1 authenticates R2 and R2 authenticates R1
|
|R1#  username R2-UID password ccie                       - R1 will authenticate R2 using R2-UID/ccie
|     !
|     interface s1/1
|      ip address 10.5.1.1 255.255.255.0
|      encapsulation ppp
|      ppp authentication pap                             - Enables R1 as the authenticator in one direction
|      ppp pap sent-username R1-UID password 0 cisco      - R1 will send this in its Auth-Req
|     !
|
|R2#  username R1-UID password cisco                      - R2 will authenticate R1 using R1-UID/cisco
|     !
|     interface s1/1
|      ip address 10.5.1.2 255.255.255.0
|      clock rate 64000
|      encapsulation ppp
|      ppp authentication pap                             - Enables R2 as the authenticator in one direction
|      ppp pap sent-username R2-UID password 0 ccie       - R2 (peer) will send this in its Auth-Req
|
>
>R1#sh users | b Interface
>       Interface    User      Mode         Idle        Peer Address
>       Se1/1        R2-UID    Sync PPP     00:00:02    10.5.1.2
>
>R2#sh users | b Interface
>       Interface    User      Mode         Idle        Peer Address
>       Se1/1        R1-UID    Sync PPP     00:00:03    10.5.1.1
```

```
*---------------------------------------------------------*
  CHAP (Challenge Handshake Authentication Protocol)
*---------------------------------------------------------*
 > CHAP uses the concept of a three-way handshake.
 > CHAP sends clear text usernames and MD5 passwords.
 > CHAP Packet Types:
     >> Challenge Packet     -     Used to begin the CHAP authentication process.
     >> Response Packet      -     Used to respond to Challenge packets.
     >> Success Packet       -     Used to indicate matching hash values.
     >> Failure Packet       -     Used to indicate non-matching hash values.

 > The Challenge and Response packets contain the following fields:
     >> Identifier           -     Incremental number used to identify a particular flow.
     >> Code                 -     Indicates packet type, either a Challenge or a Response packet.
     >> Value Field          -     The Challenge packet's value is a random number.
                                   The Response packet's value is a one-way calculated hash number.
     >> Name Field           -     The hostname of the router sending the packet.

 > After the Link Establishment Phase is complete, the authenticator sends a Challenge packet to the peer.
     >> The Challenge packet contains a new ID value, a random number and the hostname of the authenticator.
 > The peer calculates a hash value and responds with a Response packet.
     >> The ID value and the random number of the received Challenge packet are fed to the MD5 generator.
     >> The peer then looks up an entry that matches the username in the 'Name' field of the Challenge packet.
     >> That password is also fed into the MD5 hash generator, which generates a one-way hash value.
     >> This hash value is set in the 'Value' field of the Response packet before it's sent back to the authenticator.
 > Upon receipt of the Response packet the authenticator also calculates a hash value.
     >> The ID value and the random number of the sent Challenge packet are fed to the MD5 generator.
     >> The authenticator looks up an entry that matches the username in the 'Name' field of the Response packet.
     >> That password is also fed into the MD5 hash generator, which generates a one-way hash value.
 > The authenticator compares its calculated hash value to the value in the Response packet.
 > If the hash values match, the authenticator sends a Success packet otherwise it sends a Failure packet.

 > CHAP supports unidirectional (one-way) and bidirectional (two-way) authentication.
 > If two-way authentication is configured and the interface goes up and down repeatedly, it means that authentication
     in one direction is failing.


 > By default, a peer uses its router hostname to identify itself to the authenticator ('Name' field value).
     >> This can be changed with "ppp chap hostname" under the interface.
 > An interface-level CHAP hostname overwrites the router's global hostname.
 > If the same hostname is specified on both sides, the session authentication will fail by default.
     >> This is because a router ignores an Auth-Req from its own hostname.
     >> The work-around is to issue the command "no ppp chap ignoreus".


 > The passwords of the matching "username" commands as described above must match between two routers.
 > An alternative to using the "username" command is to use the interface-level password command "ppp chap password".
 > A global password is always tried first and then an interface-level password is tried.
```

```
CONFIG-SET : PPP one-way CHAP authentication
+-----------------------------------------------
|Example: R2 connects to R1, where R1 authenticates R2
|Refer to Output-101 section for debug output from this configuration
|
|R1#  username R2 password cisco                          - The UID (R2) is the hostname of the peer
|     !
|     interface s1/1
|      ip address 10.5.1.1 255.255.255.0
|      encapsulation ppp                                    - Enables PPP encapsulation
|      ppp authentication chap                              - Enables R1 as the authenticator
|
|R2#  username R1 password cisco                            - The password 'cisco' must match between R1 and R2
|     !                                                     - Alternatively  "ppp chap password" could have been used
|     interface s1/1
|      ip address 10.5.1.2 255.255.255.0
|      clock rate 64000
|      encapsulation ppp
>
>R1#sh users | b Interface
>       Interface    User     Mode         Idle       Peer Address
>       Se1/1        R2       Sync PPP     00:00:04   10.5.1.2
```

```
CONFIG-SET : PPP two-way CHAP authentication
+------------------------------------------------
|Example: R2 connects to R1
|R1 authenticates R2 and R2 authenticates R1 separately
|
|R1#  username CCIE password cisco                          - Username CCIE is the hostname send by the peer
|     !                                                     - The password must match between peers
|     interface s1/1
|       ip address 10.5.1.1 255.255.255.0
|      encapsulation ppp                                    - Enables PPP encapsulation
|      ppp authentication chap                              - Enables R1 as the authenticator in on direction
|
|
|R2#  username R1 password cisco                            - Matches the global hostname of R1
|     !                                                     - The password must match between peers
|     interface s1/1
|      ip address 10.5.1.2 255.255.255.0
|      encapsulation ppp
|      ppp authentication chap                              - Enables R2 as the authenticator in on direction
|      ppp chap hostname CCIE                               - Interface hostname overwrites routers hostname (R2)
```

```
CONFIG-SET: PPP PAP authentication one direction and PPP CHAP authentication in the other direction
+-----------------------------------------------------------------------------------------------------
|R1 authenticates R2 using CHAP and R2 authenticates R1 using PAP
|
|R1#  username R2-CHAP password cisco2                        - Username/password pair used to validate CHAP
|     !
|     interface s1/1
|      ip address 10.5.1.1 255.255.255.0
|      encapsulation ppp
|      ppp authentication chap                                - Enables R1 as the authenticator for CHAP
|      ppp pap sent-username R1-PAP password cisco1           - R1 (PAP-peer) will send this in its Auth-Req
|     !
|     !
|R2#  username R1-PAP password cisco1                         - Username/password pair used to validate PAP
|     !
|     interface s1/1
|      ip address 10.5.1.2 255.255.255.0
|      encapsulation ppp
|      ppp authentication pap                                 - Enables R2 as the authenticator for PAP
|      ppp chap hostname R2-CHAP                              - Defines the R2 hostname sent to R1
|      ppp chap password cisco2                               - Defines the CHAP password for the R2-CHAP hostname
|
```

```
*-----------------------------------------*
   EAP (Extensible Authentication Protocol)
*-----------------------------------------*
```

```
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Security Configuration Guide: Securing User Services, Release 12.4T
         > AAA (Authentication, Authorization and Accounting
          > Authentication
           > Radius EAP Support
```

```
 > EAP is a newer protocol that can also be used for PPP authentication.
 > EAP operates very similarly to CHAP:
      >> It uses a three-way handshake process.
      >> It uses mostly similar packet types.
      >> It can also make use of MD5.

 > The Advantages of EAP
      >> The EAP protocol can support multiple authentication mechanisms.
      >> The EAP implementation does not pre-negotiate a particular mechanism during LCP Phase.
      >> Instead postpones that function until the Authentication Phase, which allows more information to be requested.

 > EAP Packet Types
      >> Request packet      -    Used to begin the EAP authentication process and indicate the mechanism used.
      >> Response packet     -    Used to respond to Request packets.
      >> Success packet      -    Used to indicate successful authentication.
      >> Failure packet      -    Used to indicate authentication failure.
 > After the Link Establishment phase is complete, the authenticator sends one or more Requests to authenticate the peer.
      >> The Request packet has a 'Type' field to indicate what is being requested.
      >> Examples of Request types include Identity,  MD5-challenge, One-Time Passwords, Generic Token Card, etc.
      >> The MD5-challenge type corresponds closely to the CHAP authentication protocol.
 > The peer sends a Response packet in reply to each Request.
      >> The Response packet contains a 'Type' field, which corresponds to the 'Type' field of the sent Request packet
 > The authenticator ends the authentication phase with a Success or Failure packet.

 > The configuration of EAP is similar to that of PAP.
 > Additionally EAP configuration requires that the local database be queried appose the default being radius:
      >> Configured with "ppp eap local" under the interface.
```

```
CONFIG-SET : PPP one-way EAP authentication
+---------------------------------------------
|Example: R2 connects to R1, where R1 authenticates R2.
|
!R1#  username R2 password ccie                            - Username R2 is the hostname send by the peer
|     !
|     interface s1/1
|      ip address 10.5.1.1 255.255.255.0
|      encapsulation ppp
|      ppp authentication eap                              - Enables R1 and the EAP authenticator
|      ppp eap local                                       - Changes default behavior to queries local database instead of radius
|
|R2
|     interface s1/1
|      ip address 10.5.1.2 255.255.255.0
|      encapsulation ppp
|      ppp eap password ccie                               - Configures the password used in Response packet
|
```

```
-----------
 COMMANDS
-----------
#sh users                                            - Shows the information about the active lines
#debug ppp authentication                            - Shows the PPP authentication, username etc
#debug ppp negotiation                               - Shows the PPP negotiation process, states, phases, routes learned and MTU's

                                                     >>> PAP Authentication <<<
#username {uid} password {pwd}                       - Verifies that an incoming username and password pair are valid
#interface s0/0
 #ppp authentication pap                             - Enables PAP authentication on the authenticator
 #ppp pap refuse                                     - Disables a peer responding to PAP. Router is a peer by default
 #ppp pap sent-username {uid} password {pwd}         - Authentication request from the client side
                                                     - With PAP the interface level command overwrites the global
 #ppp max-bad-auth {number}                          - Specifies the maximum number of authentication tries
```

```
                                                        >>> CHAP Configuration <<<
#username {uid} password (pwd}                          - Username specified here needs to match remote side hostname
#interface s0/0
 #ppp authentication chap                               - Enables CHAP authentication on the authenticator
 #ppp chap hostname {uid}                               - Allows alternate CHAP hostname, instead of routers hostname
 #ppp chap password {pwd}                               - Defines an interface-specific CHAP password. Global password is tried first
 #ppp chap refuse                                       - Disables a client responding to CHAP. A router is a client by default
 #no ppp chap ignoreus                                  - Hidden command to allow both sides to have the same hostname configured
 #ppp chap splitnames                                   - Hidden command to allow different hostnames for a CHAP challenge/response
 #ppp max-bad-auth {number}                             - Specifies the maximum number of authentication tries
```

```
                                                        >>> EAP Configuration <<<
#username {uid} password (pwd}                          - Verifies that an incoming hostname and password pair are valid
#interface s0/0
 #ppp authentication eap                                - Enables EAP authentication on the authenticator
 #ppp eap local                                         - Changes default behavior to queries local database instead of radius
 #ppp eap password {pwd}                                - Configures the password used in Response packet
 #ppp eap refuse                                        - Disables a client responding to EAP. A router is a client by default
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=================================================*
   MPPE (Microsoft Point-To-Point Encryption)
*=================================================*
```

- DOC-CD LOCATION

```
       > 12.4T Configuration Guides
        > Cisco IOS VPDN Configuration Guide, Release 12.4T
         > Configuring Client-Initiated Dial-In VPDN Tunneling
          > MPPE Encryption of PPTP Tunnels
```

- Defined by RFC 3078.
- PAP, CHAP and EAP only secures(authenticates) the link between the two end-points, but not the data.
- MPPE is an older protocol for encryption over a PPP link for better security between the end-points.
- MPPE offers the use of two different types of encryption based on the size of the key.
- The two key sizes supported are 40-bit, and 128-bit and the encryption itself uses an RC4 cipher.
- There are two modes of MPPE encryption:

```
 > Stateful MPPE Encryption
      >> Provides better performance, but is badly affected by packet loss.
 > Stateless MPPE Encryption
      >> Provides lower performance, but is more tolerable in terms of packet loss.
```

- PPP encryption is negotiated during the NCP phase of PPP link negotiation by CCP (Compression Control Protocol).
- MPPE requires that MS-CHAP (Microsoft's implementation of CHAP) is configured and working beforehand.

```
CONFIG-SET: Configuring MPPE between two peers
+-------------------------------------------------------
|Example: R2 connects to R1, where R1 authenticates R2
|
|R1#  username R2 password ccie                          - Username/password used to authenticate hostname R2
|     !
|     interface s1/1
|      ip address 10.5.1.1 255.255.255.0
|      encapsulation ppp
|      ppp authentication ms-chap                        - Enables R1 as MS-CHAP authenticator
|      ppp encrypt mppe 40 required                      - Configures MPPE using 40-bit encryption
|
|
|R2#  interface s1/1
|      ip address 10.5.1.2 255.255.255.0
|      encapsulation ppp
|      ppp chap password ccie                            - Defines the password used by R2
|      ppp encrypt mppe 40                               - Configures MPPE using 40-bit encryption
|

-----------
 COMMANDS
-----------
# sh ppp mppe {int}                                - Displays the MPPE information for an interface

#interface s1/1
 #ppp authentication ms-chap                       - Enables MS-CHAP authentication on the authenticator
 #ppp encrypt mppe {auto|40|128} [passive|required] [stateful]
                                                   - Enables MPPE on the interface
                                                   - {auto|40|128} Specifies the type of encryption
                                                   - [passive] MPPE will only be negotiated if requested
                                                   - [required] MMPE must be negotiated, else link is terminated
                                                   - [stateful] MPPE will negotiate only stateful encryption


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
  PPP Reliable Link
*======================*
- Defined by RFC 1663.
- Defines a method of negotiating and using Numbered Mode LAPB to provide a reliable serial link.
- Numbered Mode LAPB provides the retransmission of error packets across the serial link.
- PPP reliable link can be used with PPP compression over the link, but does not require PPP compression to work
- PPP reliable link does not work with multilink PPP.

-----------
 COMMANDS
-----------
# sh int                                           - Will show whether LAPB has been established on the link.
# debug lapb                                       - Displays all traffic for interfaces using LAPB encapsulation.
```

```
#interface s0
 #ppp reliable-link                                       - Enables PPP reliable-link
```

-=-

LQM (Link Quality Monitoring)

- The PPP suite includes a feature that allows routers to analyze the quality of the link.
- LCP provides an optional Link Quality Determination phase. In this phase, LCP tests the link to determine whether the link quality is sufficient to use layer3 protocols.
- The command "ppp quality percentage" ensures that the link meets the quality requirement set; otherwise, the link is brought down.
- The percentages are calculated for both incoming and outgoing directions.

COMMANDS

```
# debug ppp packet                                        - Shows specific LCP operation

#interface s0
 #ppp quality {percentage}                                - Enables link quality monitoring
```

-=-

MLP (Multilink PPP)

```
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Dial Technologies Configuration Guide, Release 12.4T
         > Part 9: PPP Configuration
          > Configuring Media-Independent PPP and Multilink PPP
           > Configuring Multilink PPP
```

- MLP provides a method for spreading traffic across multiple physical WAN links while providing packet fragmentation and reassembly, proper sequencing, multivendor interoperability and load balancing on inbound and outbound traffic.
- MLP fragmentation sends the fragments simultaneously over multiple point-to-point links to the same remote address.
- MLP can measure the load on just inbound traffic or on just outbound traffic, but not on the combined load of both inbound and outbound traffic.

CONFIG-SET: MLP - Configuring a Multilink PPP Bundle

```
+-------------------------------------------------------
|     interface s0/0
|      no ip address
|      encapsulation ppp
|      ppp multilink                                    - Enables MLP on serial0/0
|      multilink-group 2                                - Assigns the interface to multilink group 2
|     !
|     interface s0/1
|      no ip address
|      encapsulation ppp
|      ppp multilink                                     - Enables MLP on serial0/1
|      multilink-group 2                                 - Assigns the interface to multilink group 2
|     !
|     interface multilink2                               - The logical options are configured on the multilink interface
|      ip address 10.5.0.1 255.255.255.0
|      ppp multilink
|      multilink-group 2
|
```

- LFI (Link Fragmentation and Interleaving)

 > Interleaving on MLP allows large packets to be multilink encapsulated and fragmented into smaller sizes to satisfy the delay requirements of real-time traffic.

 > Small real-time packets are not multilink encapsulated and are transmitted between the fragments of the large packets.

 > The interleaving feature also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be transmitted earlier than other flows.

 > Interleaving applies only to interfaces that can configure a multilink bundle interface.

 > WFQ on MLP works at the packet level, not at the level of multilink fragments.

 > IOS calculates the fragment size using the formula:
 Fragment size = max-delay * bandwidth.

- MCMP (Multiclass Multilink PPP)

 > This feature allows the delivery of delay-sensitive packets, such as the packets of a voice call, to be expedited by omitting the PPP multilink protocol header and sending the packets as raw PPP packets inbetween the fragments of larger data packets.

- MRRU Negotiation

```
 > DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Dial Technologies Configuration Guide, Release 12.4T
         > Part 9: PPP Configuration
          > PPP/MLP MRRU Negotiation Configuration
```

 > The PPP/MLP MRRU negotiation configuration feature allows a router to send and receive frames over MLP bundles that are larger than the default Maximum Receive Reconstructed Unit (MRRU) limit of 1524 bytes.

```
-----------
 COMMANDS
-----------
#interface s0/1
 #ppp multilink                                        - Enables MLP
 #multilink-group {no}                                 - Specifies interface multilink group membership
 #ppp multilink interleave                             - Enables LFI (real-time packet interleaving)
 #ppp multilink fragment-delay {ms}                    - Configures a maximum fragment delay
 #ppp multilink multiclass                             - Enables MCMP on an interface
 #ppp multilink mrru [local | remote] {mrru-value}     - Configures the MRRU value negotiated on a MLP bundle
                                                       - [local] Configures the local MRRU value
                                                       - [remote] The min value to be accepted from the peer
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=================================*
  PPPoFR (PPP over Frame-Relay)
*=================================*
 > DOC-CD LOCATION
       > 12.4T Configuration Guides
        > WAN
         > Cisco IOS Wide-Area Networking Configuration Guide, Release 12.4T
          > Part 1: Frame-Relay
           > PPP over Frame-Relay
```

- Frame-relay does not natively support features such as authentication, link quality monitoring, or reliable transmission.
- Authentication of frame-relay PVCs or binding of multiple PVCs together is made possible by implementing PPPoFR.
- PPPoFR is configured using virtual-template interfaces.
- A virtual-template is a PPP encapsulated interface that is used to spawn a 'template' of configuration down to multiple member interfaces.
- When using a virtual-template interface it's important to understand that a virtual-access 'member' interface is cloned from the virtual-template interface when the PPP connection comes up.
- Therefore the virtual-template interface itself will always be in the DOWN/DOWN state.
- This could affect certain network designs using features like the "backup interface" command.
- The virtual template number used is locally significant and does not have to match between neighbors.

CONFIG-SET: PPPoFR example using authentication

```
+---------------------------------------------------
|     interface virtual-template1                    STEP1 - Creates the virtual-template interface
|      ip address 10.5.1.1 255.255.255.0                   - Configures the logical options like an IP address on the template
|      ppp chap hostname ROUTER1                           - "encapsulation ppp" not needed (virtual-templates always run PPP)
|      ppp chap password 0 CISCO                           - Authentication is optional
|      !
|     interface s0/0                                  STEP2- Enters the physical frame-relay interface and
|      encapsulation frame-relay                               binds the virtual-template to the frame-relay PVC
|      frame interface-dlci 101 ppp virtual-template1      - Note that the order of these steps are important
|      !
>     #sh ip interface brief | include 10.5.1.1
>      virtual-template1 10.5.1.1 YES manual down down     - The virtual-template interface will always be down/down
>      virtual-access1 10.5.1.1 YES TFTP up up             - The virtual-access interface indicates the circuit status
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=================================*
  PPPoE (PPP over Ethernet)
*=================================*
- DOC-CD LOCATION
       > 12.4 T Configuration Guides
        > Cisco IOS Broadband Access Aggregation and DSL Configuration Guide, Release 12.4T
         > PPP Configuration
          > PPPoA, PPPoE, PPPox
           > PPP over Ethernet Client (client side)
        or > Providing Protocol Support for Broadband Access Aggregation of PPPoE Sessions (server side)
```

```
- PPPoE provides the ability to connect a network of hosts to a RAS (Remote Access Server) (also called a B-RAS (Broadband RAS)).
- PPPoE enables access control, billing and type-of-service to be done on a per-user basis.
- To establish a point-to-point connection over ethernet, each PPP session learns the Ethernet address of the remote peer.
- The use of PPPoE is commonly seen in DSL (Digital Subscriber Line) networks.

- Cisco IOS routers can act as a PPPoE client and/or server.
- A PPPoE session is initiated by the client.
- A dialer interface must be used for cloning virtual access.
- Remember that a virtual-template interface is a PPP interface, there is no need for "encapsulation ppp".
- It is recommended to set the MTU to 1492 bytes. 1492 bytes accommodates for the PPPoE header encapsulation of 8 bytes
      in the ethernet frame payload to avoid fragmentation.

- Dialer Persistent Feature:
 > Allows a DDR (Dial-on-Demand routing) dialer profile connection to be brought up without being triggered by interesting traffic.
 > The connection is not brought down until the shutdown interface command is entered on the dialer interface.
 > If the persistent connection is torn down for some other reason, the system immediately tries to bring the connection back up.


CONFIG-SET: Basic PPPoE config using a local IP pool
+-------------------------------------------------------------
|R1 is acting as the PPPoE client, R2 is acting as the PPPoE server
|
|R1#  interface fa0/0                                 >>> PPPoE Client Config <<<
|     interface fa0/0
|      no ip address
|      ip mtu 1492                                     - Changes the IP MTU value to avoid fragmentation
|      pppoe enable group global                       - Enables PPPoE on the interface
|      pppoe-client dial-pool-number 1                 - Adds fa0/0 to dialer pool 1
|     !
|     interface Dialer1
|      ip address negotiated                           - IP address negotiated during IPCP setup
|      encapsulation ppp                               - Non PPP interfaces needs it enabled
|      dialer pool 1                                   - Associates the dialer interface with a dialer pool
|      dialer idle-timeout 0                           - Changes the default idle timeout of 120-sec to zero
|      dialer persistent                               - Forces the dialer interface to be connected at all times
|     !
|
|R2#                                                  >>> PPPoE Server Config <<<
|     bba-group pppoe global                           - Creates a global PPPoE profile
|      virtual-template 1                              - Assigns the profile to the virtual-template interface
|     ip local pool PPP-Pool 10.5.0.1                  - Defines a IP pool to server IP addresses from
|     !
|     interface fa0/0
|      no ip address
|      ip mtu 1492                                     - IP MTU is reduced to cater for PPPoE framing overhead
|      pppoe enable group global                       - Enables PPPoE on the interface
|     !
|     interface virtual-template1
|      ip address 10.5.0.2 255.255.255.0               - Configures the server-side address
|      peer default ip address pool PPP-Pool           - Assigns an IP address from pool during IPCP phase
```

```
CONFIG-SET: Basic PPPoE config using a DHCP
+--------------------------------------------------
|R1 is acting as the PPPoE client, R2 is acting as the PPPoE server
|
|R1#  interface fa0/0                                    >>> PPPoE Client Config <<<
|      no ip address
|      ip mtu 1492                                      - Changes the IP MTU value to avoid fragmentation
|      pppoe enable group global                        - Enables PPPoE on the interface
|      pppoe-client dial-pool-number 1                  - Adds fa0/0 to dialer pool 1
|     !
|     interface Dialer1
|      ip address dhcp                                  - IP address negotiated using DHCP
|      encapsulation ppp                                - Non PPP interfaces needs it enabled
|      dialer pool 1                                    - Associates the dialer interface with a dialer pool
|      dialer idle-timeout 0                            - Changes the default idle timeout of 120-sec to zero
|      dialer persistent                                - Forces the dialer interface to be connected at all times
|     !
|
|R2#                                                     >>> PPPoE Server Config <<<
|     ip dhcp excluded-address 10.5.0.2 10.5.0.254      - Excludes a reserved/used IP addresses
|     !
|     ip dhcp pool PPP-Pool                             - Create a DHCP pool
|      network 10.5.0.0 255.255.255.0                   - Defines the DHCP prefix
|     !
|     bba-group pppoe global                            - Creates a global PPPoE profile
|      virtual-template 1                               - Assigns the profile to the virtual-template interface
|     !
|     !
|     interface fa0/0
|      no ip address
|      ip mtu 1492                                      - IP MTU is reduced to cater for PPPoE framing overhead
|      pppoe enable group global                        - Enables PPPoE on the interface
|     !
|     interface Virtual-Template1
|      ip address 10.5.0.2 255.255.255.0                - Configures the server-side address
|      peer default ip address dhcp-pool PPP-Pool       - Assigns an IP address from pool during IPCP phase
|
```

```
-----------
 COMMANDS
-----------
# show vpdn session                                  - Shows the PPPoE interfaces, states, and MAC addresses used

# clear vpdn tunnel pppoe                            - Terminates PPPoE session and immediately try to re-establish the session
# clear interface dialer {number}                    - With dialer persistent, re-attempts to bring up the connection

# debug vpdn pppoe-data                              - Displays PPPoE session data packets
# debug vpdn pppoe-errors                            - Displays errors preventing a session establishment and terminating errors
# debug vpdn pppoe-events                             - Displays PPPoE session establishment events messages
# debug vpdn pppoe-packets                            - Displays each PPPoE protocol packet exchanged
# debug dialer                                        - Displays info about the packets received on a dialer interface

#dialer-list {group} protocol {prot} list {acl}       - References ACL listing interesting traffic or use the persistent command
#bba-group pppoe {name}                               - Creates a PPPoE profile
 #virtual-template {number}                           - Assigns the profile to the virtual-template interface

#interface fa1/1
 #pppoe-client dial-pool-number 1                     - Associates fa1/1 to dialer pool number 1

#interface dialer0
 #mtu 1492                                            - Adjusts MTU size to accommodate PPPoE framing overhead
 #ip address negotiated                               - Specifies the IP address to be obtained from the PPPoE server
 #dialer pool {number}                                - Associates the dialer interface with a dialer pool
 #dialer persistent [delay sec] | [max-attempts]      - Forces a dialer interface to be connected at all times
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  PPP Half-Bridging
*=====================*
- DOC-CD LOCATION
       > 12.2 Mainline Configuration Guides
        > Cisco IOS Dial Technologies Configuration Guide, Release 12.2
         > PPP Configuration
          > Configuring Media-Independent PPP and Multilink PPP
            > Configuring PPP Half-Bridging
```

- When a serial interface is configured as a PPP half-bridge, the link to the remote bridge functions as a virtual ethernet interface, with the serial interface functioning as a node on that remote network.
- When a packet is received by the PPP half-bridge, it is converted to a routed packet and forwarded normally.
- The reverse process happens for packets destined for the remote bridge.
- An interface cannot function as both a half-bridge and a bridge.

```
-----------
 COMMANDS
-----------
#interface Ethernet0
 #ppp bridge ip                                       - Enables PPP half-bridging for IP (must be done before configuring the IP)
 #encapsulation ppp
 #ip address 10.1.1.2 255.0.0.0                       - Provides a protocol address on the same subnetwork as the remote network.
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   Troubleshooting PPP
*=========================*
- When troubleshooting PPP link establishments, consider the following:
 > For back-to-back serial interfaces running PPP:
      >> Which end is the DCE and which side is the DTE?                                        # sh controllers {int} | i DCE|DTE
      >> Is the DCE end configured to provide clocking?                                         # sh run int {int} | i clock rate
 > Is the physical interface connected and unshut?                                              # sh int {int}
 > Is PPP encapsulation configured on both ends?                                                # sh run | i interface|encap.*ppp
 > Is the PPP enabled interface showing the LCP in OPEN state ?                                 # sh int {int} | i LCP
 > If needed was an IP address negotiated successfully?                                         # sh ip int brief | i IPCP
 > For other LCP/NCP problems run a debug and consider debug output below.                      # debug ppp negotiation


- When troubleshooting PPP authentication, consider the following:
 > PPP authentication does not begin until the LCP phase is complete and is in an OPEN state.   # sh int {int} | i LCP
 > PPP authentication issues are almost always configuration errors!
 > If two-way authentication was configured, is the interface going up, down, up, down, etc.?   # sh logg | i {int}
      >>  Authentication in one direction is working but failing in the opposite direction.
 > For PAP authentication issues:
      >> Confirm PAP authentication is configured correctly.                                    # sh run | i pap|username
      >> Is the PAP server-side configured as the authenticator?                                # sh run | i auth.*pap
      >> Do the usernames and passwords match between peers?                                    # sh run | i username
      >> If not analyze the debug output to see what the cause of failure could be.             # debug ppp authentication
 > For CHAP authentication issues:
      >> Confirm CHAP authentication is configured correctly.                                   # sh run | i chap|username
      >> Do the passwords match between peers?                                                  # sh run | i password
      >> Does the local router's hostname match the peer's username command?                    # sh run | i username
            >>> If needed, are the neighbors allowed to use the same hostname?                  # sh run | i ignoreus
      >> If not analyze the debug output to see what the cause of failure could be.             # debug ppp authentication
 > For EAP authentication issues:
      >> Confirm EAP authentication is configured correctly.                                    # sh run | i eap|username
      >> Do the passwords match between peers?                                                  # sh run | i password
      >> Does the local router's hostname match the peer's username command?                    # sh run | i username
      >> If not analyze the debug output to see what the cause of failure could be.             # debug ppp authentication
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===============*
   OUTPUT-101
*===============*
----------------------------------------
      Simple PPP negotiation
----------------------------------------
- Example debug output from the PPP negotiation process using "debug ppp negotiation".
- R1-to-R2 using PPP encapsulation.
 > Only interface IP addresses are configured.
 > No PPP features configured other than default enabled features.

* %LINK-3-UPDOWN: Int Serial1/1, changed state to up              - Link state changes to UP
* Se1/1 PPP: Using default call direction
* Se1/1 PPP: Treating connection as a dedicated line
* Se1/1 PPP: Session handle[36000002] Session id[2]
* Se1/1 PPP: Phase is ESTABLISHING, Active Open                   - Changes to Link Establishment Phase
                                                                  - I = Incoming, and O = Outgoing
* Se1/1 LCP: O CONFREQ [Closed] id 2 len 10                       - LCP is initialized, Conf-Req send out
* Se1/1 LCP:    MagicNumber 0x03244884 (0x050603244884)                 - The Magic number is used for loop prevention
* Se1/1 LCP: I CONFREQ [REQsent] id 12 len 10                     - Configuration request received inbound
* Se1/1 LCP:    MagicNumber 0x042447B0 (0x0506042447B0)
* Se1/1 LCP: O CONFACK [REQsent] id 12 len 10                     - ID-12 acknowledged by sending a Conf-Ack
* Se1/1 LCP:    MagicNumber 0x042447B0 (0x0506042447B0)
* Se1/1 LCP: I CONFACK [ACKsent] id 2 len 10                      - Received Conf-Ack, peer acknowledged ID-2
* Se1/1 LCP:    MagicNumber 0x03244884 (0x050603244884)
* Se1/1 LCP: State is Open                                        - LCP was successful, hence the LCP state is OPEN
* Se1/1 PPP: Phase is FORWARDING, Attempting Forward
* Se1/1 PPP: Phase is ESTABLISHING, Finish LCP
* Se1/1 PPP: Phase is UP                                          - NCP is initiated and IPCP started
* Se1/1 IPCP: O CONFREQ [Closed] id 1 len 10                      - IPCP Conf-Req send to peer
* Se1/1 IPCP:    Address 10.5.1.1 (0x03060A050101)                - This is the local interface IP advertised
* Se1/1 IPCP: I CONFREQ [REQsent] id 1 len 10                     - Received Conf-Req from peer
* Se1/1 IPCP:    Address 10.5.1.2 (0x03060A050102)                - This is the peer interface address
* Se1/1 IPCP: O CONFACK [REQsent] id 1 len 10                     - Local router acknowledges all IPCP parameters
* Se1/1 IPCP:    Address 10.5.1.2 (0x03060A050102)                - Local router acknowledges the peer's address
* Se1/1 IPCP: I CONFACK [ACKsent] id 1 len 10                     - Peer router acknowledges all IPCP parameters
* Se1/1 IPCP:    Address 10.5.1.1 (0x03060A050101)
* Se1/1 CDPCP: O CONFREQ [Closed] id 1 len 4                      - CDP negotiation take place
* Se1/1 CDPCP: I CONFREQ [REQsent] id 1 len 4
* Se1/1 CDPCP: O CONFACK [REQsent] id 1 len 4
* Se1/1 IPCP: State is Open                                       - IPCP NCP phase successful, state changed to open
* Se1/1 CDPCP: I CONFACK [ACKsent] id 1 len 4
* Se1/1 CDPCP: State is Open                                      - CDP negotiated successfully
* Se1/1 IPCP: Install route to 10.5.1.2                           - Peer neighbor route installed by default
* %LINEPROTO-5-UPDOWN: Line protocol on Serial1/1, changed state to up
                                                                  - Line-protocol state changes to UP
```

```
---------------------------------------------
      One-way PAP authentication
---------------------------------------------
- Example debug output of a successful PPP one-way PAP authentication.
- Output using "debug ppp authentication" is done on the authenticator.

* %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
* Se1/1 PPP: Using default call direction
* Se1/1 PPP: Treating connection as a dedicated line
* Se1/1 PPP: Session handle[17000004] Session id[5]
* Se1/1 LCP:    AuthProto PAP (0x0304C023)                        - Authentication negotiated during the LCP phase
!
!-----Not shown here is the full LCP negotiations
!                                                                 - I = Incoming, and O = Outgoing
* Se1/1 PPP: Authorization required                               - After LCP state is open, Authentication phase begins
* Se1/1 PAP: I Auth-Req id 3 len 16 from "R2-UID"                 - Authentication receives a Auth-Req from the peer
* Se1/1 PAP: Authenticating peer R2-UID                           - Verifies received username matches a local username
* Se1/1 PPP: Sent PAP LOGIN Request
* Se1/1 PPP: Received LOGIN Response PASS                          - Username/password pair match is successful
* Se1/1 PPP: Sent LCP AUTHOR Request
* Se1/1 PPP: Sent IPCP AUTHOR Request
* Se1/1 LCP: Received AAA AUTHOR Response PASS
* Se1/1 IPCP: Received AAA AUTHOR Response PASS
* Se1/1 PAP: O Auth-Ack id 3 len 5                                - Authenticator acknowledges authentication as successful
!
!-----Not shown here is the IPCP negotiations
!
* %LINEPROTO-5-UPDOWN: Line protocol on Serial1/1, changed state to up
```

```
---------------------------------------------
      One-way CHAP authentication
---------------------------------------------
- Example debug output of a successful PPP one-way CHAP authentication.
- Output using "debug ppp authentication" is done on the authenticator.


* %LINK-3-UPDOWN: Interface Serial1/1, changed state to up        - Physical link state changes to UP
!
!-----Not shown here is the full LCP negotiations
!                                                                 - I = Incoming, and O = Outgoing
* Se1/1 PPP: Authorization required                               - After LCP state is open, Authentication phase begins
* Se1/1 CHAP: O CHALLENGE id 1 len 23 from "R1"                   - The authenticator sends a Challenge packet to the peer
* Se1/1 CHAP: I RESPONSE id 1 len 23 from "R2"                    - Response to ID-1 received back from the peer
* Se1/1 PPP: Sent CHAP LOGIN Request
* Se1/1 PPP: Received LOGIN Response PASS                          - Indicates local and received hash values match
* Se1/1 PPP: Sent LCP AUTHOR Request
* Se1/1 PPP: Sent IPCP AUTHOR Request
* Se1/1 LCP: Received AAA AUTHOR Response PASS
* Se1/1 IPCP: Received AAA AUTHOR Response PASS
* Se1/1 CHAP: O SUCCESS id 1 len 4                          - Authenticator acknowledges authentication as successful
!
!-----Not shown here is the IPCP negotiations
!
* %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up



-----------------------------------------
      Additional debug outputs
-----------------------------------------
- These are examples of different possible outputs from DIFFERENT debug sessions
- The output will be from either "debug ppp negotiation" or  "debug ppp authentication"
- Copy descretly owned by Kane  Bagwell


* PPP: Phase is AUTHENTICATING, by both                     - This indicates two-way authentication

* PPP: Phase is AUTHENTICATING, by this end                 - This indicates one-way authentication
                                                            - The local router is the authenticator

* PPP: Phase is AUTHENTICATING, by the peer                 - This indicates one-way authentication
                                                            - The local router is the peer

* PAP: O Auth-Nak id 23 len 26 msg is "Authentication failed"     - Hostname/Password mismatch. Seen on authenticator

* CHAP: Unable to authenticate for peer                     - The host supplied by peer is not configured locally

* PPP: Received LOGIN Response FAIL                         - Local and received hash values do not match
* CHAP: O FAILURE id 49 len 25 msg is "Authentication failed"     - This error indicates a password mismatch

* PPP: Received LOGIN Response FAIL                         - Same problem, but different version of IOS output
* CHAP: O FAILURE id 16 len 25 msg is "MD/DES compare failed"     - This error indicates a password mismatch
```

```
 _  ______      ______                   _
| | |  ___ \   |  ___ \              _  (_)
| | | |___) )  | |___) )___  _  _ _| |_ _ ____   ____
| | |  _____/  |  __  //  _ \| | | (_   _) |  _ \ /  _  |
| | | |        | |  \ | |_| | |_| | | | |_| | | | ( (_| |
|_| |_|        |_|   \_\___/|_____/  \__)_|_| |_|\___  |
                                                  (___|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-*
|      INDEX            |
*-=-=-=-=-=-=-=-=-=-=-=-*
- Routing Decisions
      + Longest Match
      + Distance
      + Inner Protocol
      + Metric
- Switching Paths
      + Process Switching
      + Fast Switching
      + CEF Switching
- Default Routing
- Floating Static Routes
- ODR (On-Demand Routing)
- Secondary IP addresses
- Backup Interface
- GRE Tunneling
- PBR (Policy-Based Routing)
- /31 Mask
- IP-Unnumbered
- Route-Maps
- Redistribution
      + Overview
            o Rules
            o Default Metrics
            o Connected Interfaces
            o Mutual Routers
      + Connected/Static Redistribution
      + RIP Redistribution
      + EIGRP Redistribution
            o Composite Metric
            o External EIGRP Routes
      + OSPF Redistribution
            o Route-Types
            o Match Keyword
      + BGP Redistribution
            o Redistribute Internal
- OER/PfR
      + Master Controller
      + Border Routers
      + Interface Types
      + Traffic-Class States
+ 5 Phases
      o Phase 1 - Profile Phase (BRs)
      o Phase 2 - Measure Phase (BRs)
      o Phase 3 - Apply Policy Phase (MC)
      o Phase 4 - Control/Enforce Phase (BRs)
      o Phase 5 - Verify Phase (MC)
```

-=-

Routing Decisions

- Refer to the following link for a flow-chart: http://routing-bits.com/2010/01/07/rib-route-selection/

- Route selection process to install routes in the RIB (Routing Information Base):

1st - Longest match/prefix

2nd - AD (Administrative Distance):

0	- Connected
1	- Static
5	- EIGRP Summary Route
20	- eBGP
90	- EIGRP
100	- IGRP
110	- OSPF
115	- IS-IS
120	- RIP
160	- ODR
170	- EIGRP external route
200	- iBGP
255	- unknown

3rd - Lowest Metric

- Exceptions to above
  > If two protocols have the same AD (if one was changed) and the router needs to decide which is best, the router will use the default AD as the tie-breaker.
  > You CANNOT have two best routes from different protocols installed into the RIB.
  > If a tie exists between OSPF routes, then O > O*IA > E1 > E2.
  > If a tie exists between BGP routes, then the BGP bestpath selection process will be followed.

- The default AD values can be changed with the "distance" command, but note differences for each protocol:

```
 > Generic         - "distance {distance}"
 > EIGRP           - "distance eigrp internal-distance external-distance".
 > OSPF            - "distance ospf {external} {inter-area} {intra-area}".
```

 COMMANDS

```
#int fa0/0
 #no routing dynamic                               - Flags the interface to indicate that routing updates should not be sent from it
                                                   - Enabled by default
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  Switching Paths
*=====================*
- There are three steps to forwarding a packet through a router:
 > Routing, finding the next hop and the exit interface.
 > Switching path, switching the packet across the backplane.
 > Finding the layer2 address.

- Only processed traffic can be debugged.
- Local traffic (destination or source being the router) is always processed switched.

- Process Switching
 > Every packet in the flow is processed by the CPU.
 > Local traffic (destination or source being the router) is always processed switched.
 > Enabled Process Switching by disabling fast switching and CEF switching at the interface level, using the commands
      "no ip route-cache" and "no ip cef".

- Fast Switching
 > With fast switching the first packet in a flow is still copied to the CPU for the layer3 lookup and housekeeping,
       before being rewritten with the layer2 destination address.
 > The switching of the first packet by the central CPU gives the CPU the opportunity to build a cache called the fast-switching
      cache, which is used to switch all subsequent packets for the same destination using the same switching path
      across the router.
 > With fast switching the cache is only built on demand, which can be timeconsuming when huge numbers of potential
       destinations are involved.
 > To avoid this, a pre-build cache was needed and thus CEF was born.

- CEF (Cisco Express Forwarding) has two main data structures:
 > The Adjacency Table:
      >> Responsible for the MAC or layer2 rewrite.
      >> This adjacency can be built from ATM, frame-relay map statements, dynamic information learned from
            ethernet-ARP, inverse-ARP on ATM, or inverse-ARP on frame-relay.
      >> The layer2 rewrite string contains the new layer2 header, which is used on the forwarded frame.
      >> For ethernet, this is the new destination and source MAC address and the ethertype.
      >> For PPP, the layer2 header is the complete PPP header, including the layer3 protocol ID.
 > FIB (Forwarding Information Base) Table
      >> The CEF table/FIB table holds the essential information, taken from the routing table, to be able to make a forwarding
            decision for a received IP packet.
      >> This information includes the IP prefix, the recursively evaluated next hop and the outgoing interface.
 > The CEF Process Flow
      >> When a packet enters the router, the router strips off the layer2 information.
      >> The router looks up the destination IP address in the CEF table (FIB) and makes a forwarding decision.
      >> The result of this forwarding decision points to one adjacency entry in the adjacency table.
      >> The information retrieved from the adjacency table is the layer2 rewrite string, which enables the router to put a new
            layer2 header onto the frame.
      >> The packet is switched out onto the outgoing interface toward the next hop.
```

- When an IP address local to the router is pinged:
 > The packet does NOT cross the backplane between interfaces. The router actually sends the packet out of the interface.
 > So, if the local interface to a peer is down, the ping will be unsuccessful.

```
----------
 COMMANDS
----------
# sh adjacency [detail]                              - Shows the layer2 adjacency table
                                                     - [detail] Optionally displays the layer2 rewrite string
# sh interface switching                             - Shows the number of packets being process-switched
# sh ip cef [prefix]                                 - Shows a CEF table entry
# sh ip cef internal                                 - Shows load-balancing hash algorithm table
# debug arp                                          - Shows the ARP queries and responses
# debug ip routing                                   - Shows the protocols and routes install/removed from the routing table
# debug ip policy                                    - Shows any policy routing information
# debug frame-relay packet                           - Shows the layer2 DLCI mapping
                                                     - 'encaps failed- no map entry': incorrect DLCI assignment
# debug ip packet [acl]                              - Shows the source, destination, exit interface, switching method
                                                     - 'unroutable': Means no route to the destination exist
                                                     - 'encapsulation failed': Means layer2 resolution is not available


#ip cef                                              - Enables CEF globally
#interface s1/0
 #no ip route-cache                                  - Disables fast-switching, thereby enabling process switching
 #ip route-cache flow                                - Enables NetFlow
 #ip load-sharing per-packet                         - Enables per-packet CEF load-balancing
#[no] ip proxy-arp                                   - Disables/Enables proxy-arp, respond with the interface MAC to hosts
                                                          destined on other networks (default = enabled)
#ip local-proxy-arp                                  - Enables the local-proxy-arp feature (requires proxy-arp enabled)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
```

Default Routing

```
*=====================*
```

- Syntax:

```
      #ip route 0.0.0.0 0.0.0.0 {next-hop IP | exit-interface}
```

- If a next-hop IP is used, it must be able to recurse to an exit interface existing in the routing table.
- If the next-hop IP is on a multipoint interface, layer3 to layer2 resolution must be configured.
- If the next-hop IP is on a point-to-point interface, layer3 to layer2 resolution is NOT required.
- An exitinterface is generally used on broadcast mediums or point-to-point links.

- IP Default-Gateway
 > Will only be used when IP-routing is disabled (useful on a switch).

- IP Default-Network
 > Network flagged as default in routing advertisements.
 > Must be a classful network that is not directly connected.

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Floating Static
*=====================*
- Example:
      #ip route 10.1.4.0 255.255.255.0 10.1.3.1 90          - Primary route
      #ip route 10.1.4.0 255.255.255.0 10.1.2.1 95          - Backup route with a higher AD than the default AD value

- Keep in mind that local interface status does not indicate end-to-end transport, especially on multipoint interfaces.
- Provided that a recursive lookup provides end-to-end next-hop reachability, the information above would work.
- If there is no end-to-end next-hop reachability, it could create a black hole.
- Assume the first route above is going across a frame-relay multipoint interface with multiple DLCIs configured.
- If the DLCI for 10.1.4.0 fails on the primary remote end but the other local DLCIs stay up, the local interface would still
      appear as UP/UP, creating a black hole. Enhanced object tracking (IP SLA) could be used as a remedy.

----------
 COMMANDS
----------
# sh ip sla monitor statistics                        - Shows brief status: UP,DOWN, last error
# sh ip sla monitor collection-statistics             - Shows detailed stats, i.e. successful, disconnects, timeouts, busy, errors
# sh track {number}                                   - Shows the configured track statement

#ip sla monitor 1                                     - Creates an RTR/IP SLA
 #type echo protocol IpIcmpEcho 10.1.3.1 [source]     - Creates a ping to destination
 #timeout 200                                         - Configures the timeout in milliseconds
 #frequency 5                                         - configures the frequency in seconds
#ip sla monitor schedule 1 start now                  - Starts running the SLA now

#track 5 rtr 1                                        - Creates a track that calls RTR (Response Time Reporter) / IP SLA monitor
#ip route 10.1.4.0 255.255.255.0 10.1.3.1 90 track 5  - Primary route: If object-1 of track-5 goes down, this route will be removed.
#ip route 10.1.4.0 255.255.255.0 10.1.2.1 95          - Backup route


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   On-Demand Routing
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > Cisco IOS IP Routing: ODR Configuration Guide, Release 12.4T

- Uses an admin distance of 160.
- Uses CDP to discover/advertise the directly connected network to the 'hub' router.
- The 'hub' router advertises a default route to the 'stub' routers via CDP.
- The stubs would respond with their connected network via CDP.
- If a stub is running an IGP, it won't respond back with its routes.
- Since CDP is disabled by default frame-relay, it must be enabled to support ODR.
 > Remember to enable CDP for the PVC it is running on, either the sub-interface or interface.
- The timers for ODR and CDP are the same.
```

COMMANDS

```
hub# sh cdp neighbors                                  - Confirms the expected stubs are showing as CDP neighbors
hub(config)#router odr                                 - Enables ODR on the HUB router
```

Secondary IP Addresses

- The big nasty that shows poor address space planning.
- Keep in mind that most traffic generated by the router out of an interface will have the primary IP as a source, not the secondary. RIP is one exception.

- Routing protocols deal differently with secondary IP addresses.
  > RIP - Can exchange updates with secondary IPs.
  > EIGRP - Can't establish neighbors using secondary IPs.
  > OSPF - Can't establish neighbors on secondary IPs, secondary networks are seen as stub-networks.

Backup Interface

- Tracks the local line protocol of the 'primary' interface.
  > If the line is up, the 'backup' interface is in standby mode.
  > If the line is down, the 'backup' interface is out of standby and UP.
- The command "backup interface {int}" is placed on the primary interface, specifying the backup interface.

- Delay-Timers can be used with the backup command.
  > 'Failover' specifies the delay before the standby link gets brought up after the primary link fails.
  > 'Failback' specifies the delay after the primary link comes back up before bringing down the secondary interface.

- This solution could have the same black-hole pitfall that floating statics have.
- One solution is to use a tunnel interface and configure the backup command on the tunnel.
- The backup command cannot be used on frame-relay physical interfaces (no way to detect when back up).

COMMANDS

```
# sh backup                                            - Shows the interfaces and the status
# debug backup                                         - Shows the backup process events

#interface {primary interface}
 #backup {backup interface}                            - Configures one interface to backup another
 #backup delay {failover - sec} {failback - sec}       - {Failover} The delay before bringing standby interface up
                                                       - {Failback} The delay after the primary came back up
```

GRE Tunneling

- GRE (Generic Route Encapsulation) is a layer3 VPN technology.
- It uses IP transport protocol 47.
- It is used to transport payload protocols over IPv4 networks.
- GRE is payload independent. It supports both IPv4 and IPv6.
- By default a GRE tunnel interface will be UP/UP before the far-end of the tunnel is configured. Enable "keepalives" to make the tunnel end-points stateful
- The GRE tunnel destination must never recurse to the tunnel interface itself, else the router will log the following error: %TUN-5-RECURDOWN:Tunnel0.

CONFIG-SET: Example GRE config on one side

```
+-----------------------------------------------
|     #interface tunnel 0
|      #tunnel source 10.1.0.1
|      #tunnel destination 10.1.0.3
|      #keepalive {period} {retries}                  - Period: How often to send keepalives. (default = 10sec)
|                                                     - Retries: Number of retry keepalives before the tunnel is brought down
```

PBR (Policy-Based Routing)

```
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Routing: Protocol-Independent Configuration Guide, Release 12.4T
        > Configuring IP Routing Protocol-Independent Features
         > Enabling Policy-Based Routing
```

- PBR (Policy-Based Routing) is a more flexible mechanism for routing packets than destination-based routing is.
- PBR allows control of traffic flow based on:

> Source/Destination
> Protocol type
> Incoming interface

- Traffic that is denied by the policy-map will get routed normally.
- By default PBR traffic is process-switched. Fast switching can be enabled with "ip route-cache policy" (see below).

CONFIG-SET: Local Policy Routing for Local Router Traffic to 'RE-ENTER' the router

```
+-----------------------------------------------------------------------------------
|     ip access-list extended LOCAL_TRAFFIC
|      permit tcp any any eq 23                             - Match locally generated telnet traffic
|     !
|     route-map LOCAL_POLICY 10
|      match ip address LOCAL_TRAFFIC                       - Redirect local telnet traffic via the loopback interface
|      set interface Loopback0                              - Traffic sent to loopback interface re-enters the router
|     !
|     interface Loopback0
|      ip address 10.5.0.1 255.255.255.0
|     !
|     ip local policy route-map LOCAL_POLICY                - Apply the policy for router generated traffic
|
```

```
----------
 COMMANDS
----------
# sh route-map                                         - Shows the configured route-maps
# debug ip policy                                      - Shows any policy routing information

#route-map {tag} [permit | deny] [sequence]            - Defines a route map to control where packets are output
 #match {options}                                      - Matches the specific match options above
 #set {options}                                        - Sets options as above

#ip local-policy route-map [route-map]                 - Applies to all traffic locally generated by the router
#ip route-cache policy                                 - Enables fast switching for policy routing
#interface S0/1
 #ip policy route-map [route-map]                      - Applies to all traffic coming into the interface on which applied
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  /31 Mask
*=====================*
- 31-bit prefixes were designed for point-to-point links.
- One bit is reserved for the host-id portion allowing only two IP addresses.
- Normally a host-id of all zeros is used to represent the network or subnet and a host-id of all ones is used to represent
      a directed broadcast. Using 31-bit prefixes, the host-id of 0 represents one host and a host-id of 1 represents the
      other host of a point-to-point link.

- Local link broadcasts (255.255.255.255) are still used with 31-bit prefixes.
- Directed broadcasts however are not possible to a 31-bit prefix. This is not really a problem because most routing protocols use
      multicast, limited broadcasts or unicasts.
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  IP-Unnumbered
*=====================*
- Allows IP processing to be enabled on a serial interface without assigning an explicit IP address to the interface.
- Should only be used on point-to-point (non-multi-access) interfaces.
- Designed to save IP addresses on point-to-point links.
- How does the routing work?
 > A router receiving a routing update installs the source address of the update as the next hop in its routing table.
 > Normally, the next hop is a directly-connected network node, but not with IP-unnumbered as the IP was 'borrowed'.
 > Instead routes learned through the IP-unnumbered interface have the interface as the next hop instead of the source address
       of the routing update.
```

```
CONFIG-SET: IP-Unnumbered Example
+-----------------------------------------
|      int eth1/0
|       ip address 10.5.0.254 255.255.255.0
|      !
|      interface ser0/0
|       ip unnumbered ethernet 0                               - Configures serial0 to 'borrow' ethernet0 IP address
|      !
|      !
>      #show ip interface brief
>       Interface     IP-Address      OK?    Method   Status   Protocol
>       Ethernet0     10.5.0.254   YES    manual   up       up
>       Serial0       10.5.0.254   YES    manual   up       up
```

```
-----------
 COMMANDS
-----------
#ip unnumbered {interface}                                 - Configures an interface to 'borrow' an IP address from another interface
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
   Route-maps
*==================================*
```

- Route-maps can be intimidating if the (if/then/set) logic behind them is not understood.
- Route-maps are processed sequentially according to the sequence numbers (default or defined), one instance at a time.
- To match all packets, simply omit the match command.

```
CONFIG-SET : Route-Map Logic
+-----------------------------------
|      route-map NAME 10                                       - Instance 10
|       match ip address 3                                     - IF the ACL matches
|       set metric 50                                          - THEN set
|      !
|      route-map NAME 20                                       - Otherwise look at instance 20
|       set metric 20
|
```

- When a route is matched against a route-map instance:
 > If the instance has a 'permit' parameter, the route will be redistributed.
 > If the instance has a 'deny' parameter, the route is not redistributed and that route is not processed further.

- If the route is not matched at all in a redistribution route-map, the route is not redistributed (implicit deny at the end).

```
- Possible match criteria and commands for redistribution:
 > Looks at the outgoing interface             #match interface {interface}
 > Using an ACL                                #match ip address {access-list-name/number}
 > Looks at the prefix and length              #match ip address prefix-list {name}
 > Based on the route's next-hop address         #match ip next-hop {ACL}
 > Matching route metrics exactly                #match metric {value}
 > Matching route metrics within range           #match metric {value} +-{deviation}
 > Protocol route type                           #match route-type {in|ex|type}
 > Matching previous defined tags                #match tag {value}


- Possible set criteria for redistribution:
 > Set the protocol route type                   #set metric-type {in|ex|type}
 > Define the destination database               #set level {stub|backbone}
 > Set the route's metric                        #set metric {value}
 > Set the unit's tag value                      #set tag tag-value



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================*
   Redistribution
*==================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Routing: Protocol-Independent Configuration Guide, Release 12.4T
        > Configuring IP Routing Protocol-Independent Features
         > IP Routing Protocol-Independent Configuration Examples

- Redistribution Rules and Guidelines:
 > Redistributed routes cannot be redistributed again on the same router!! (RIP > OSPF >  EIGRP)
 > Manual Split-Horizon - Never redistribute a prefix injected from domain-A into domain-B back to domain-A.
 > You cannot change the EIGRP external AD (170) per-route. It can only be done for all prefixes or none.
 > Sub-optimal routing in the lab is not a problem, unless specified, as long as there is full reachability.
 > The redistribute command redistributes only routes that are in the router's current IP routing table (RIB).
 > Before enabling any redistribution, make sure each protocol has full reachability within itself.
 > The metric assigned using the "redistribute metric" command takes precedence over metrics assigned with the
     "default-metric" command.

- Redistributing into RIP and EIGRP
 > The metrics must be set via configuration as RIP and EIGRP have no default values.
 > RIP cannot use a 0 metric, the hop count must be between 1 and 16.
 > The 0 metric is also incompatible with the EIGRP multi-metric format.

- Redistributing into OSPF
 > By default, routes are redistributed into OSPF as external type-2 (E2) routes, with a metric of 20.


- These logical steps happen when redistribution is enabled:
 STEP 1
      >> The router looks ONLY at the routing table to get the routes that are to be redistributed.
      >> Not all the routes that the redistributed protocol sends to the routing table, will be redistributed.
      >> Verify the routes with "sh ip route <redistributed protocol>" before redistribution is enabled.
```

STEP 2

>> The router takes all connected subnets matched by that routing protocol’s network commands.

>> Verify these interfaces by looking at the redistributed protocol's network statements OR look at the individual routes with "sh ip route x.x.x.x" as listed with "advertised by".

>> Passive-interfaces for the redistributed protocol ARE included when redistributing.

>> This hidden step is the equivalent of the following (Hidden: H>):

```
#router ospf 1
 #redistribute rip subnets

 H>redistribute connected subnets route-map NAME
H>route-map NAME permit
 H>match interface fa0/0 s0/1                    - All the RIP enabled interfaces
```

>> If ever asked to redistribute specific interfaces, ALWAYS INCLUDE the interfaces that the redistributed protocol runs on.

- Mutual Router Redistribution

> Redistributing from a low AD protocol(e.g. OSPF) to a higher AD protocol(e.g. RIP) won’t cause feedback as the lower AD is always preferred.

> But redistributing from a high AD protocol to a low AD protocol could create problems, because the high AD protocol routes might prefer the redistributed low AD routes to a destination.

> Four ways to correct possible route feedback:

>> On one of the redistributing routers, increase the AD for routes redistributed from the low AD protocol (e.g. OSPF) to an AD higher, say 130, than the HIGH AD protocol (e.g. RIP).

>> Filter the redistributed high AD routes into the origination protocol to stop them feeding back via the low AD protocol.

>> Use tags to filter out redistributed high AD routes with a specific tag.

>> Route-summarization could also be used.

> Always protect the higher AD protocol(e.g. RIP) and tag the routes from the higher AD protocol to be filtered.

RIP -> OSPF - Tag upon redistribution into OSPF.
OSPF -> RIP - Filter the tagged routes to stop them going back into RIP.

> If two protocols have the same AD (e.g. if one was changed) and the router needs to decide which is best, the router will use the default AD as the tie-breaker.

RIP

- RIP uses the following logical two step process used for redistribution.
 > Firstly: "sh ip route rip", will advertise ONLY the RIP route in the routing table to a neighbor.
 > Secondly: all the connected interfaces running RIP will be advertised.
- So when other protocols are redistributed on a RIP router an onwards RIP stub router behind it might not get all the RIP routes, that were received by other RIP routers, since the redistributed protocol's routes might be preferred, due to the AD.
- In short, RIP cannot advertise RIP routes if they are not in the routing table.
- When redistributing any protocol into RIP, the metric must be specified.
- RIP can't interpret a zero metric and only uses a metric(hop-count) value between 1 and 15.

- This can be done by
  > Specifying the 'metric' keyword with the redistribution command.
  > Using the "default-metric" command for RIP.
  > Setting the metic in a route-map and referencing it in the redistribution command.

EIGRP

- With EIGRP and OSPF mutual redistribution, by default there won't be any feedback issues because:
  > OSPF routes have an AD of 110.
  > EIGRP routes have an AD of 90.
  > EIGRP redistributed routes have an AD of 170.
- Problems arise when there are external EIGRP routes (D EX) in the EIGRP domain with an AD of 170.
- When these get redistributed into OSPF and back into EIGRP, traffic to the external routes from the EIGRP domain originally could prefer a path via OSPF with an AD of 110, causing a loop.

!!NOTE!! Before redistributing EIGRP into any protocol, check the routing table for 'D EX' routes.

- The feedback can be fixed with:
  > Distance command.
  > Tag filtering.
  > Matching only certain OSPF route types for redistribution.

- It is also advisable to specify a meaningful tag when redistributing routes into EIGRP. This can help one to see where the route was redistributed, e.g. use a tag like 3120 where 3=router3 and 120=OSPF-AD.

- The distance for external EIGRP routes (D EX) CANNOT be changed on a per-route basis, changes can only be done per route-type, i.e. for ALL external EIGRP routes(D EX).
- EIGRP requires the "no auto-summary" command, otherwise classful subnets will be redistributed.

OSPF

- By default, external IGP routes are redistributed into OSPF as external type 2 (E2) routes with a cost of 20.
- External BGP routes are also redistributed into OSPF as external type 2 (E2) routes, but with a cost of 1.
- The 'subnets' keyword is a requirement with OSPF otherwise only classful network addresses are redistributed.
- It is also advisable to specify a meaningful tag when redistributing routes into OSPF.

- OSPF External Type1 (E1) Routes
  > Include the external cost as well as the internal cost to the ASBR.
  > Used to exit the AS as close as possible to the destination.
  > Mostly used if multiple exit points exist for an AS.

- OSPF External Type2 (E2) Routes
  > Include only the external cost of the route.
  > Used to exit the AS via closest ASBR.
  > Often used with only one OSPF exit point.

```
- Order of route preference among OSPF routes (O > O*IA > E1 > E2):
 1- Intra-area OSPF            > O
 2- Inter-area OSPF routes     > O*IA
 3- External OSPF E1 routes    > E1
 4- External OSPF E2 routes    > E2

- The 'routing bit set' field from "sh ip ospf database" means the OSPF is sending the route to the routing table.
 > Whether it is installed depends on the presence, or not, of better routes.

- Redistributing OSPF into any protocol gives you the option to redistribute only certain OSPF route types
      with the 'match' keyword.


*----------------------*
      BGP
*----------------------*
- Redistributing OSPF into BGP
 > By default, if the 'match' keyword is not defined BGP will redistribute only the route type OSPF INTERNAL.

- Redistributing BGP into any other protocol
 > Generally not advised in production networks.
 > Only eBGP-learned prefixes are redistributed into the IGP.
 > By default iBGP-learned prefixes are NOT candidates for redistribution. This is a blackhole safeguard.
 > This can be disabled by using the command "bgp redistribute-internal".

- BGP routes originated through the "network" command have an origin code of 'i-igp'.
- BGP routes originated through redistribution have an origin code of '?-incomplete'.


 ----------
 COMMANDS
 ----------
# sh ip route profile                                  - Shows rapid/constant route changes, useful when looping occurs
# sh ip protocol                                       - Useful to verify RIP routes, routing-sources and timers
# sh ip ospf database                                  - Useful to see which router advertised an OSPF route

 #redistribute connected [metric] [route-map]          - Redistributes connected interfaces into a protocol
 #redistribute static [metric] [route-map]             - Redistributes static routes into a protocol

#router rip                                            >>> REDISTRIBUTING RIP <<<
 #redistribute {protocol} [metric] [transparent] [route-map]
                                                       - Redistributes other routes into RIP
                                                       - [metric]: RIP metric is hop count (value 1-16)
                                                       - [transparent]: Allows BGP to carry the redistributed RIP metric
                                                             Commonly used in MPLS
 #distance {ad} {src-ip} {wildcard} [acl]              - Changes the AD for all RIP routes received from the source router
                                                       - [ACL] Could be used to match only certain routes
 #default-metric {value}                               - Sets the default metric for all redistributed routes
```

```
#router eigrp {asn}                                  >>> REDISTRIBUTING EIGRP <<<
 #redistribute {protocol} metric {bw} {dly} {rely} {load} {mtu} [route-map] [tag]
                                                     - Redistributes other routes into EIGRP
 #default-metric {bw} {dly} {rely} {load} {mtu}      - Sets the default metric for redistributed routes
 #distance eigrp {internal} {external}               - Changes the AD for ALL internal and external EIGRP routes

#router ospf {pid}                                   >>> REDISTRIBUTING OSPF <<<
 #redistribute {protocol} [subnets] [metric] [metric-type 1|2] [tag] [route-map]
                                                     - [Subnets] : Without this keyword only major network addresses
                                                         are redistributed
 #neighbor {ip} cost {metric}                        - Specifies cost for a specific neighbor
                                                     - Useful for NBMA network when preferring one DLCI
 #default-metric {metric}                            - Sets the default metric for redistributed routes
 #distance ospf {external} {inter-area} {intra-area} - Changes the AD for external, inter/intra-area OSPF routes


#router bgp {ans}                                    >>> REDISTRIBUTING BGP <<<
 #redistribute {igp} [pid] [metric] [route-map] [subnets]
                                                     - Redistributes other routes into BGP
 #distribute-list {acl1} out {igp}                   - Filters routes redistributed from specified routing process
 #bgp redistribute-internal                          - Allows the redistribution of iBGP learned routes into a IGP
                                                          (default = only eBGP routes)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
    OER/PfR
*======================*
```

```
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > Cisco IOS Optimized Edge Routing Configuration Guide, Release 12.4T
```

- Traditional routing uses static metrics and destination-based prefix reachability.
- Traditional network recovery is primarily based on neighbor and link failures.

- Deploying OER/PfR enables intelligent network traffic load distribution and dynamic failure detection for data paths at the network edge.

- OER/PfR monitors traffic class performance and selects the best entrance or exit for traffic classes.
- Adaptive routing adjustments are based on RTT, jitter, packet-loss, path availability, traffic load and cost.

- PfR (Performance Routing) is the successor of OER (Optimized Edge Routing).
- OER provided route control on a per-destination prefix basis.
- PfR expands these capabilities and in addition facilitates intelligent route control on a per-application basis.

- There is minimal CPU impact using OER/PfR but it does utilize a lot more memory, which is based directly on the amount of prefixes.
- The Master Controller has the biggest impact.

```
- An OER/PfR deployment has two primary components, a master controller and one or more border routers.
- Both of these functions could be configured on the same router, e.g. one router with two exit interfaces; the must be at minimum
    two interfaces to exit the local autonomous system.

- MC (Master Controller)
 > Monitors the network and maintains a central policy database with statistical information.
 > Makes all policy decisions and controls the BRs.
 > Maintains communication and authenticates the sessions with the BRs using MD5.
 > MC will not become active if there are no BRs or only one exit point exists.
 > The MC compares long-term (60 min) and short-term (5 min) measurements.
 > Then applies default or user-defined policies to alter routing to optimize prefixes and exit links.
 > Can support up to 10 BRs and up to 20 OER-managed external interfaces.
 > Does not have to be in the forwarding path, but must be reachable by BRs.

- BRs (Border Routers)
 > Edge routers with one or more exit links to an ISP or another WAN.
 > Report prefix and exit link measurements to the MC.
 > Enforce policy changes from the MC, by injecting preferred routes to alter routing in the network.
 > The preferred route can be an injected BGP route or an injected static route.
 > BRs must be in the forwarding path.
 > OER BRs must use outbound next hops that are on different subnets.

- Internal Interfaces
 > Interfaces between the MC and the BRs.
 > Used for OER communication and for passive monitoring.
 > At least one internal interface connecting to the inside network is required per BR.

- External Interfaces
 > Used to forward outbound traffic from the network.
 > Used as the source for active monitoring.
 > At least two external interfaces are required in an OER-managed network.

- Local Interface
 > The source for communications between the BRs and the MC.
 > A loopback interface could be used for this.
 > If both the MC and BR functions are configured on the same router then a loopback interface should be used.

- OER communication between the MC and the BRs is carried separately from routing protocol traffic.
- IOS 12.4(9)T introduces the ability to monitor and control inbound traffic.
- Prefixes or traffic classes pass through different states after they are learned.
- Traffic-Class States:
 > Default         - Not under OER control, but routed based on existing routing (prefixes start out in this state).
 > Choose          - The MC is choosing an exit link (don't blink or you may miss this state).
 > Holddown        - The MC moved the prefix to a new exit. No policy changes are applied while the prefix is in a
                       holddown state.  This is intended to prevent flapping.
 > In-Policy       - The status of the prefix matches the policies. No changes are made in this state until the config
                       or performance measurements change.
 > Out-of-Policy   - The prefix does not match any policy. Active probing or passive monitoring (or both) will be used to
                       find a better exit while the prefix is in this state. If none are found the MC will use the
                       best one available.
```

- There are five OER phases:
 > Phase 1 - Profile Phase (BRs)
 > Phase 2 - Measure Phase (BRs)
 > Phase 3 - Apply Policy Phase (MC)
 > Phase 4 - Control/Enforce Phase (BRs)
 > Phase 5 - Verify Phase (MC)

Phase 1 - Profile Phase

 > The list of traffic-class entries is called an MTC-list (Monitored Traffic Class list).
 > The entries in the MTC-list can be profiled either by automatically learning the traffic flows or by manually configuring the traffic classes.
 > Both methods can be used at the same time.
 > BRs profile interesting traffic, a function which has to be optimized by learning the flows as they pass through the router.
 > Non-interesting traffic will be ignored.
 > BRs sort traffic based on delay and throughput and send it to the MC.

 > Automatic learning can be done in three ways:
    1- Prefix traffic class
        >>> The OER MC can be configured, using the NetFlow top talker functionality, to automatically learn prefixes based on the highest outbound throughput or the highest delay time.
        >>> Performance measurements for the prefix-based traffic classes are reported to the MC where the learned prefixes are stored in the MTC list.
        >>> All incoming and outgoing traffic flows are monitored. The top 100 flows are learned by default, but this can be changed.
        >>> The MC can be configured to aggregate learned prefixes based on type, BGP or non-BGP (static)
        >>> Prefixes can be aggregated based on the prefix length (default = /24).

    2- Application traffic class learning
        >>> In addition, layer4 options such as protocol or port numbers can be used to identify specific application traffic classes.
        >>> DSCP values are also supported.

    3- Learn list config mode
        >>> Learn lists are a way to categorize learned traffic classes.
        >>> In each learn list, different criteria including prefixes, application definitions, filters and aggregation parameters for learning traffic classes can be configured.

 > Manual learning can be done in two ways:
    1- Manual prefix traffic class configuration
    2- Manual application traffic class configuration

Phase 2 - Measure Phase

 > The network has to measure the performance metrics of the previous created individual traffic classes.
 > OER automatically configures (virtual) IP SLA probes (ICMP by default) and NetFlow configurations.
 > No explicit IP SLAs or NetFlow configurations are required.

```
> OER also measures the utilization of the links.
> By default, all traffic classes are passively monitored using the integrated NetFlow functionality.
> OOP (Out-Of-Policy) traffic classes are actively monitored using IP SLA functionality (learned probe).
> OER measures the performance of both traffic classes and links but, before monitoring a traffic class or link, OER checks the
     state of the traffic class or link (refer to the traffic-class states above).
>  After determining the state of the traffic class or link OER may initiate one of the following performance measuring modes:
     >> Passive monitoring
           >>> Looks at actual traffic, utilizing NetFlow data statistics as traffic traverses a router.
           >>> Measures the following metrics:
                 >>>> Delay          - Based on TCP RTT (Round Trip Time) (initial SYN to the following ACK).
                 >>>> Packet loss   - Tracks TCP sequence numbers for each TCP flow.
                 >>>> Reachability - Tracks TCP-SYNs that weren't acknowledged with TCP-ACKs.
                 >>>> Throughput    - by measuring the total number of bytes and packets for non-TCP traffic flows.

     >> Active monitoring
           >>> Generates synthetic traffic to emulate the traffic class that is being monitored (using IP SLA probes).
           >>> Measures the probes with the following metrics:
                 >>>> Delay          - Based on TCP RTT (initial SYN to following ACK).
                 >>>> Reachability - Tracks TCP-SYNs that didn't receive TCP-ACKs.
                 >>>> Jitter         - Measures the variable delay between packets arriving at the destination.
                 >>>> MOS            - Standards-based method of measuring voice quality.
                 >>>> Learned probes (ICMP) are automatically generated when a traffic class is learned using NetFlow.

     >> Both active and passive
           >>> Combining both active and passive monitoring in order to generate a more complete picture of traffic flows.

> Fast Failover
     >> Could be enabled, all exits are continuously probed using active monitoring and passive monitoring.
     >> Probe frequency can be set to a lower frequency than other methods.
     >> Allows faster failover capability, i.e. failover can occur within 3 sec.


*-----------------------------------*
    Phase 3 - Apply Policy Phase
*-----------------------------------*
> By default, OER runs in an observe mode during the Profile, Measure and Apply Policy phases (no changes to the network are made
     until the OER is configured to control the traffic).

> After collecting the performance metrics, OER compares the results with a set of configured low and high thresholds for
     each metric.

> Policies define the criteria for determining an out-of-profile event.
> There are two types of policies that can be defined:
     >> Traffic class policies     - Defined for prefixes or for applications.
     >> Link policies              - Defined for exit or entrance links at the network edge (overwrites traffic policies).

> An OER policy is a rule that defines an objective and contains the following attributes:
     >> A scope                    - The network traffic sent to the specific traffic class entry.
     >> An action                  - A routing table change.
     >> A triggered event          - The violation of a measured threshold.
```

```
> Link grouping introduces a method of specifying preferred links for one or more traffic classes in an OER policy, so that the
     traffic classes are routed through the best link from a list of preferred links, referred to as the primary link group.
> A fallback link group can also be specified in case there are no links in the primary group that satisfy the specified
     policy and performance requirements.

> Three types of mode options are available in a policy:
     >> Mode monitor {active | passive | both}
     >> Mode route {control | metric | observe}
     >> Mode select-exit {best | good}

> Three types of timers can be configured as OER policy operational parameters:
     >> Backoff timer
           >>> Adjust the transition period for which the MC holds an out-of-policy traffic class entry.
           >>> MC will wait for the transition period before making an attempt to find an in-policy exit.

     >> Holddown timer
           >>> The minimum period of time that a new exit must be used for before an alternate exit can be selected.
           >>> Used to prevent the traffic class entry from flapping because of rapid state changes.

     >> Periodic timer
           >>> The MC tries to find a better path for a traffic class entry, even if the traffic class entry is
                 in-policy on the current exit.

> Policies may conflict- one exit point may provide the best delay while the other has the lowest link utilization.
> A 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.
> The variance configures the acceptable range (%) of deviation from the best metric among all network exits.


*--------------------------------------*
    Phase 4 - Control/Enforce Phase
*--------------------------------------*
> In this phase the traffic is controlled to enhance the network performance time.
> OER will initiate route changes when one of the following occurs:
     >> A traffic-class goes OOP.
     >> An exit link goes OOP.
     >> The periodic timer expires and the 'select exit' mode is configured as 'select best' mode.
> A measured prefix's parent route, with a valid next-hop, must exist before a new prefix will be injected (this could
     be a default route).
> OER exit link selection control techniques on BRs depend on the routing setup with the internal/external network:
     1- BGP Peering
           >>> BGP is used to peer internally and externally.
           >>> When eBGP is used with the outside autonomous systems, the local preference attribute can used to set a higher
                 preference for injected routes.

     2- BGP redistribution into an IGP
           >>> BGP is used to the ISP and an IGP (OSPF, EIGRP, RIP) is used internally.
           >>> The BRs should advertise a single, default route to the internal network (IGPs).
```

```
      3- Static route and/or redistribution into an IGP
            >>> Used in a network where only static routing is configured, then no redistribution is required.
            >>> Or used in a network where an IGP is deployed and static routes to the border router exit interfaces are
                  configured. These static routes must be redistributed into the IGP.
            >>> If need be OER alters routing for this type of network by injecting temporary static routes.
            >>> The temporary static route replaces the parent static route.
            >>> OER will not inject a temporary static route unless a parent static route doesn't exist.
            >>> OER applies a default tag value of 5000 to identify the injected static route.
            >>> To avoid routing loops, the redistributed OER static routes should never be advertised over a WAN by an OER
                   border router or any other router.

 > OER entrance link selection control techniques:
      1- BGP autonomous system number prepend
            >>> After OER selects a best entrance for an inside prefix, extra AS hops (up to a maximum of six) are prepended
                  to the other inside BGP prefix advertisements over the other entrances.
            >>> This will make the best entrance a more preferred entry point.
            >>> Copy descretly owned by Kane  Bagwell
      2- BGP autonomous system number community prepend
            >>> After OER selects a best entrance for an inside prefix, a BGP prepend community can be attached to the
                   inside prefix.

*--------------------------------*
    Phase 5 - Verify Phase
*--------------------------------*
 > 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.
 > OER uses NetFlow to automatically verify the route control.
 > If the traffic class is still OOP the previous optimizing changes will be reverted to.

- This does look like a mouthful, but read it two or three times and it won't seem so bad.
- The easiest to understand, is to see most of it put together, Look at the following config-set.
- Here is one MC and two BRs.

CONFIG-SET: Configuring OER/PfR with auto-learning and control options
+---------------------------------------------------------------------
>>MC CONFIGURATION
|
|     key-chain KEY1
|      key 1
|       key-string pfr                                      - This defines the key-chain to be used later
|     !
|     oer master
|      logging                                              - Enables sysloging
|      mode route control                                   - Enables the MC to make control decisions
|      prefixes 1000                                        - Learn and monitor a 1000 routes during learning period
|      backoff 90 3000 300                                  - Sets time periods (min, max and step) for policy decisions
|      learn
|       delay                                               - Enables learning based on the highest delay time
|       monitor period 8                                    - The amount of time the router will learn prefixes
|       periodic interval 15                                - The time between the learning periods
|       !
|      border 10.5.100.1 key-chain KEY1                - Defines the first BR and authentication
|       interface fa0/0 internal                       - Specifies the BR1 internal interface
|       interface s0/0 external                        - Specifies the BR1 external interface
|        max-xmit-utilization absolute 1500            - Specifies the outbound traffic to 1.5MB
|        cost-minimization fixed fee 1000              - Assigns a cost value making this interface more preferred
|      !
|      border 10.5.104.1 key-chain KEY1                - Defines the second BR and authentication
|       interface fa0/0 internal                       - Specifies the BR2 internal interface
|       interface s0/0 external                        - Specifies the BR2 external interface
|        max-xmit-utilization absolute 1000            - Specifies the outbound traffic to 1MB
|        cost-minimization fixed fee 800               - Assigns a cost value making this interface less preferred
|
>>BR1 CONFIGURATION
|
|     key-chain KEY1
|      key 1
|       key-string pfr
|     !
|     oer border
|      master 10.5.10.1 key-chain KEY1
|      local fa0/0                                      - Defines the local interface used for communication
|      active-probe address source int fa0/0           - This BR will source active probes from fa0/0
|
>>BR2 CONFIGURATION
|
|     key-chain KEY1
|      key 1
|       key-string pfr
|     !
|     oer border
|      master 10.5.10.1 key-chain KEY1
|      local fa0/0
|      active-probe address source int fa0/0           - This BR will source active probes from fa0/0
|
```

```
----------
 COMMANDS
----------
# sh oer master                           - Shows traffic classes, aggregation, filters, key list etc
# sh oer master policy                    - Shows policy information, i.e. timers, next-hop etc
# sh oer master prefix                    - Shows the status of the monitored prefixes
# sh oer master link-group                - Shows information about configured OER link groups
# sh oer master traffic-class             - Shows information about traffic classes that are monitored and controlled
# sh oer border                           - Shows detailed info about the BR and connecting MC
# sh oer border passive learn             - Shows traffic class filter and aggregation ACL information
# sh oer border passive cache             - Shows real-time prefix information collected from the BR
# sh oer border passive prefixes          - Shows the passive measurement information collected by NetFlow
# sh oer border active-probes             - Shows connection status, info about active probes
# sh oer border routes {bgp | static}     - Shows information about OER controlled routes
# sh ip cache verbose flow                           - From the BR will display all the flows, protocols, ports, etc
# debug oer border routes {bgp | static | [detail]}  - Used to debug parent route lookup and route changes


#no oer master                                       - Disables a MC and completely remove the process config
#oer master
 #shutdown                                           - Temporarily disables a MC and stops an active MC process


#no oer border                                       - Disables a BR and completely remove the process config
#oer border
 #shutdown                                           - Temporarily disables a BR and stops an active BR process


#key chain {C-NAME}                                  >>> CONFIGURING THE KEY-CHAIN <<<
 #key {KEY-ID}                                       - Identifies an authentication key on a key chain
  #key-string {TEXT}                                 - Specifies the authentication string for the key


#oer master                                          >>> CONFIGURING THE MC <<<
 #border {ip} [key-chain {C-NAME}]                   - Establishes communication with the 1st BR
  #interface {interface} {internal | external}       - Specifies the BR interface as an OER-managed internal or external
 #border {ip} [key-chain {C-NAME}]                   - Establishes communication with the 2nd BR
  #interface {interface} {internal | external}       - Specifies the BR interface as an OER-managed internal or external
 #port {number}                                      - (o) Changes the default port 3949 for communication between the MC and BRs
 #logging                                            - (o) Enables syslog messages for a MC or BRs process
 #keepalive {timer}                                  - (o) Change the OER keepalive time, (def = 60 sec)


#oer border                                          >>> CONFIGURING THE BR <<<
 #master {ip} [key-chain {C-NAME}]                   - Enters the MC IP address and key-chain to establish communication
 #local {INTERFACE}                                  - Identifies a local interface
 #port {NUMBER}                                      - (o) Changes the default port 3949 for communication between the MC and BRs


#oer master                                          >>> AUTOMATIC LEARNING <<<
 #learn                                              - Enters OER top talker and top delay learning config mode
  #delay                                             - Enables prefix learning based on the highest delay time
  #throughput                                        - Enables prefix learning based on the highest outbound throughput
  #inside bgp                                        - Enables inside prefixes learning
  #protocol {protocol | tcp-port | udp-port}         - Enables prefix learning based on protocol and port numbers
  #traffic-class keys {default | [sport | dport | dscp | prot]}
                                                     - Defines the fields used when learning prefixes
  #traffic-class filter access-list {ACL}            - Enables filtering of class when using passive monitoring
  #aggregation-type {bgp | non-bgp | prefix-length}  - (o) Aggregate learned prefixes based on traffic flow type
  #monitor-period {min}                              - (o) The time that a MC learns traffic flows (def = 5min)
  #periodic-interval {min}                           - (o) The time interval between prefix learning periods (def = 120min)
  #prefixes {number}                                 - (o) The number of prefixes learn during monitoring periods (def = 100)
  #expire after {session number | time minutes}      - (o) How long learned prefixes are kept in the central policy database
```

```
                                                      >> MANUAL PREFIX CONFIGURATION <<<
#ip prefix-list {NAME} [seq] {deny|permit} [le]       - Creates an prefix-list to manually select prefixes for monitoring
#oer-map {M-NAME} {sequence}                          - Enters OER map config mode
 #match ip address prefix-list {NAME}                 - References the prefix-list (only one match statement allowed per oer-map)


                                                      >> MANUAL APPLICATION PREFIX CONFIGURATION<<<
#ip access list {standard | extended} {NAME}          - Creates an access-list to manually select prefixes for monitoring
 #permit {tcp|udp} {src}[port] {dst}[port] [dscp]     - Sets conditions to match protocol, TCP/UDP port number or DSCP
#oer-map {M-NAME} {sequence}                          - Enters OER map config mode
 #match ip address access-list {ACL}                  - References a ACL (only one match statement allowed per oer-map)


#oer master                                           >>> CONFIGURING ROUTE MONITORING <<<
 #mode monitor {active | both | passive}              - Sets route monitoring or route control mode {default = both}
 #mode monitor fast                                   - Enables fast failover, using active and passive monitoring, 3 sec failover
 #max-range-utilization percent {value}               - Sets the maximum utilization range for all OER-managed exit links
 #max range receive percent {value}                   - Sets the upper limit of the receive utilization for entrance links
 #border ....                                         - Enters BR config-mode
  #interface ... external                              - Enters interface mode
   #max-xmit-utilization {absolute kbps | percentage} - Modifies the OER exit (outbound) link utilization threshold
   #maximum utilization receive {absolute | percent}  - Modifies the OER entrance (inbound) link utilization threshold
   #cost-minimization {calc| discard| end| fixed fee} - Configures cost-based optimization policies
 #active-probe {echo| tcp-conn target-port| udp-echo target-port}
                                                      - Configures an active probe for a target prefix
 #active-probe address source interface {interface}   - Configures the source address of an active probe
#ip sla monitor responder                             - Enables remote device to respond to IP SLA probes


#oer master                                           >>> SETTING INDIVIDUAL POLICY PARAMETERS <<<
 #backoff {min-timer} {max-timer} [step-timer]        - (o) Used to adjust the time period for policy decisions
 #periodic {timer}                                    - (o) Sets OER to periodically select the best exit link
 #holddown {timer}                                    - (o) Sets the traffic-class entry-route dampening timer
 #delay {relative {average} | threshold {maximum}     - Sets the delay threshold (If exceeded, the prefix is out-of-policy)
                                                      - {relative} sets a percentage of loss based on a comparison of short-term
                                                            and long-term packet loss percentages.
                                                      - {threshold} sets the absolute packet loss based on packets per million
 #loss {relative {average} | threshold {maximum}}     - Sets the packet loss limit that OER will permit for a traffic class entry
                                                      - {relative} sets a percentage of loss based on a comparison of short-term
                                                            and long-term packet loss percentages.
                                                      - {threshold} sets the absolute packet loss based on packets per million
 #unreachable {relative {average} | threshold {maximum}}- Sets the maximum number of unreachable hosts
 #resolve {cost priority | range priority | delay {priority|variance} | loss {priority|variance} | utilization {priority|variance}}
                                                      - Sets policy priority or resolves policy conflicts.
 #mode select-exit {best | good}                      - Enables the exit link selection based on performance or policy
                                                      - {best} Selects the best available exit
                                                      - {good} Selects the first in-policy exit
```

```
#oer-map {M-NAME} {sequence-number}                >>> SETTING UP A POLICY-MAP <<<
 #match ip address {access-list | prefix-list}     - References a ACL or IP prefix-list as match criteria (only match allowed)
 #match oer learn {delay | inside | throughput}    - Specifies how to match OER learned prefixes for optimization
 #set backoff {min-timer} {max-timer} [step-timer] - (o) Used to adjust the time period for policy decisions
 #set periodic {timer}                             - (o) Sets OER to periodically select the best exit link
 #set holddown {timer}                             - (o) Sets the traffic-class entry-route dampening timer
 #set delay {relative {average} | threshold {maximum} - Sets the delay threshold (If exceeded, the prefix is out-of-policy)
                                                   - {relative} sets a percentage of loss based on a comparison of short-term
                                                         and long-term packet loss percentages.
                                                   - {threshold} sets the absolute packet loss based on packets per million
 #set loss {relative {average} | threshold {maximum}} - Sets the packet loss limit that OER will permit for a traffic class entry
                                                   - {relative} sets a percentage of loss based on a comparison of short-term
                                                         and long-term packet loss percentages.
                                                   - {threshold} sets the absolute packet loss based on packets per million
 #set resolve {cost priority | range priority | delay {priority|variance} | loss {pri|var} | utilization {pri|var}}
                                                   - Sets policy priority or resolves policy conflicts.
 #set unreachable {relative {average} | threshold {maximum}}
                                                   - Sets the maximum number of unreachable hosts
 #set jitter {threshold maximum}                   - Configures the jitter threshold value
 #set mos {threshold {minimum} | percent {percent}} - Configures the MOS threshold and percentage values
 #set mode select-exit {best | good}               - Enables the exit link selection based on performance or policy
                                                   - {best} Selects the best available exit
                                                   - {good} Selects the first in-policy exit


#oer master                                        >>> LINK GROUPING <<<
 #border ....                                      - Enters BR config-mode
  #interface ... external                          - Enters interface mode
   #link-group {link-group-name}                   - Configures a border router exit interface as a member of a link group


#oer master                                        >>> CONFIGURING EXIT POLICY CONTROL <<<
 #mode route control                               - Enables route control mode, to dynamically implements change if needed
 #mode route metric bgp local-pref {value}         - Sets a BGP local preference value for injected BGP routes
 #mode route metric static tag {value}             - Sets a static tag value for injected static routes


#oer master                                        >>> CONFIGURE INBOUND POLICY CONTROL <<<
 #mode select-exit best                            - Configures exit selection settings
 #no resolve delay                                 - Disables any priority for delay performance policies
 #no resolve loss                                  - Disables any priority for loss performance policies
 #max range receive percent {percentage }          - Sets the percentage difference between the inbound traffic utilizations
 #border ....                                      - Enters BR config-mode
  #interface ... external                          - Enters interface mode
   #maximum utilization receive {absolute | percent} - Sets the maximum inbound traffic utilization per interface
   #downgrade bgp community {value}                - Sets the downgrade options for BGP advertisement
 #oer-map {MNAME} {sequence-number}                - Enters OER map config mode
  #match oer learn {delay | inside | throughput}   - A match clause entry in an OER map to match OER learned prefixes
  #set delay {relative | threshold}                - Creates a set clause entry to configure the delay threshold
  #set mode route control                          - Creates a set clause entry to configure route control for matched traffic
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY

```
 ______    _    ______
|  ___ \  | |  |  ___ \
| |___) ) | |  | |___) )
|  __  /  | |  |  _____/
| |  \ \  | |  | |
|_|   \_| |_|  |_|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-*
|       INDEX           |
*-=-=-=-=-=-=-=-=-=-=-=-*
```

```
- RIP Operation
      + Output-Delay
- Metrics and Timers
- RIP Version 1 and 2
- Update Types
      + Broadcast
      + Multicast
      + Unicast
- Network Statement
- Passive Interface
- Split-Horizon, RIP Triggered
- Summarization
- Filtering
      + Distribute-List
      + Offset List
      + Distance
- Default Routing
- Authentication
      + MD5
      + Text
- Troubleshooting RIP
      + RIP Updates
      + Route-Selection Issues
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================*
   RIP Operation
*====================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Routing: RIP Configuration Guide, Release 12.4T
        > Configuring Routing Information Protocol

- All RIP messages are encapsulated in UDP with source/destination ports being 520.
- Two message types:
 > Request  - Used to ask neighboring routers to send an update.
 > Response - Carries the update/routing entries.
```

- If a router needs to send an update with more than 24 route entries, multiple RIP messages will be produced.
- If more than one route exists to the same destination with equal hop counts, equal-cost load balancing will be performed.
- RIP sees secondary IP addresses on interfaces as separate data links and can exchange routes with a secondary IP.
- If there are multiple addresses on an interface, RIP will send an update packet sourced with each RIP-enabled range on an interface.
- RIP performs a source-validation check, where the source IP address of incoming routing updates must be on the same IP network as one of the addresses defined for the receiving interface.
- The source-validation may need to disabled when the source address is on a different subnet from the locally configured address, i.e. local has a /32 and remote side has a /24. This can be seen with "debug ip rip events".

- Output-delay
  > Used to set an inter-packet gap between 8 and 50msec (default=0).
  > Can be used when a high-speed router is sending updates to a low-speed router.

COMMANDS

```
# ping 224.0.0.9                                      - RIPv2 enabled routers will answer the ping and respond
# debug ip rip events                                 - Shows RIP protocol events

#router rip                                           - Enables the RIP process
 #no validate-update source                           - Disables the validation of the source address in updates
 #output-delay {ms}                                   - Sets an inter-packet gap (value 8-50) (default=0)
```

-=-

=========================

Metrics and Timers

=========================

- RIP uses hop-count as a metric.
- 1 hop per interface.
- 16 hops = unreachable.

```
- Update timer (30 sec)       - A router sends a response message out of every RIP-enabled interface every 30 sec on average.
- Invalid timer (180 sec)     - Amount of time a route can stay in the routing table without being updated.
- Holddown (180 sec)          - An update with a hop count higher than the metric recorded in the table will place
                                     a route in holddown.
- Flush timer (240 sec)       - The time after which an invalid route gets removed from the routing table.
                              - Before the flush timer expires, the invalid route will be advertised with the unreachable metric.
                              - Shows in the routing table as "x.x.x.x is possibly down"
```

COMMANDS

```
#timers basic {update} {invalid} {holddown} {flush}   - Changes the default RIP timers
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
   RIP Version 1 and 2
*==========================*
- By default, a RIP process configured on a Cisco router sends only RIPv1 messages but listens to both RIPv1 and RIPv2 messages.
- The version 2 command causes RIP to send and listen to RIPv2 messages only.

- RIPv2 is version 1 with the following extensions:
 > Subnet masks carried with each route entry.
 > Authentication of routing updates.
 > Next-hop addresses carried with each route entry.
 > External route tags.
 > Multicast route updates.



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================*
   Update Types
*==================*
- Broadcast Updates
 > Are the default for RIPv1.
 > With RIPv2 broadcast updates are optional.
 > Broadcast at an interface level for RIPv2 is enabled with "ip rip v2-broadcast".

- Multicast Updates
 > Are the default for RIPv2 using 224.0.0.9.
 > RIPv2 is enabled with "version 2".

- Unicast Updates
 > Are optional for both RIPv1 and RIPv2.
 > Are enabled using "neighbor {ip}" under the RIP process.
 > Are useful on NBMA networks such as frame-relay.
 > Unicast updates do not stop the sending of broad/multicasts packets. To achieve that use "passive-interface".



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   Network Statement
*======================*
- A network statement on RIP has no mask option and assumes classful boundaries, even with RIPv2.
- The updates sent to neighbors use the assigned subnet masks from the interface on which the 'network' address is configured.

-----------
 COMMANDS
-----------
#router rip
 #network {ip}                                         - Specifies matched interfaces to be advertised by RIP
```

Passive interface

- "Passive-interface" is not a RIP-specific command.
- "Passive-interface" stops the sending of updates (Response.Msg) out of the interface specified.
- The router will still listen to RIP updates and update its routing table accordingly upon receipt of a response update message on the passive interface.
- The router will still advertise that interface address in normal updates to other peers.
- To stop the transmission of broad/multicast updates and send only unicast updates to a neighbor, include the "passive-interface" command along with the "neighbor" command under the RIP process.

COMMANDS

```
#router rip
 #passive-interface default                         - Disables sending of RIP updates on all interfaces
 #[no] passive-interface {interface}                - Stops the sending of updates out of the interface specified
                                                    - Still receives updates and populates the routing table
                                                    - Still advertises that interface in normal updates to other peers
```

Split-Horizon, RIP Triggered

- RIP employs split-horizon with poison reverse and triggered updates.
- Copy descretly owned by Kane Bagwell

- Split-Horizon

> Updates received in an interface will not be sent out of the same interface.
> Might be undesirable on partial mesh NBMA networks such as multipoint interfaces.
> Is enabled for all interfaces by default, except main physical interfaces in frame-relay, which have it disabled by default.

- RIP Triggered

> A triggered update occurs whenever the metric for a route is changed and, unlike regularly scheduled updates, includes only the entries that have changed.
> The receiving router does not reset its update timer when a triggered update is received.
> The command "ip rip triggered" enables the triggered extensions of RIP. It is needed on both sides of a link.
> Route table updates are minimized to include only the initial exchange of route tables and updates when changes to the route tables occur.
> The triggered state goes from DOWN, through INIT and LOADING, to FULL.
> Should only be configured on a point-to-point serial link.

COMMANDS

```
#interface s0/0
 #ip rip triggered                                  - Enabled triggered updates
                                                    - Only available on serial links, if both sides are enabled
 #no ip split-horizon                               - Disables split-horizon
```

Summarization

- By default auto-summarization is enabled for RIP.
- Limitation of RIP summarization:
  > More than one major network summary per interface is not allowed.
  > Cannot summarize past the major network. For example, a summary of 10.0.0.0/7 is not allowed.

- When doing manual summarization make sure auto-summary is off.
- The defining characteristic of a classful routing protocol is that it does not advertise an address mask along with the advertised destination address.

- For every packet passing through the router:
  1- If the destination address is a member of a directly connected major network, the subnet mask configured on the interface attached to that network will be used to determine the subnet of the destination address. Therefore, the same subnet mask must be used consistently throughout that major network.
  2- If the destination address is not a member of a directly connected major network, the router will try to match only the major class A, B, or C portion of the destination address.

COMMANDS

```
#interface fa0/0
 #ip summary-address rip {ip} {mask}                  - Limits the advertisements out of that interface to ONLY the summary
                                                      - A subnet of the aggregate must be in the RIP database
```

Filtering

- RIP can use distribute-lists, offset-lists and the "distance" command to filter traffic.
- Inbound filtering can be source based, like the distribute-list/ACL example below.
- If the subnet mask must be matched, rather use prefix-lists.

- A "offset-list" can be used to modify the metric, but only to increase the metric. The metric cannot be decreased.
- A "offset-list" can also be used to filter traffic, by setting the metric to unreachable.
- Access-list '0' matches all routes.
- If no interface is identified, the list will modify all incoming or outgoing updates specified by the access list on any interface.
- If no access-list is called (by using a zero as the access list number), the offset list will modify all incoming or outgoing updates.

```
CONFIG-SET: RIP Offset-List Example
+-----------------------------------
|     access-list 1 permit 10.5.0.0 0.0.255.255              - Identifies the route entry for subnet 10.5.0.0/16
|     !
|     router rip
|      network 10.0.0.0                                      - For routes coming in on serial0,
|      offset-list 1 in 2 Serial0                                 matching the ACL-1, add 2 hops to the metric
|


CONFIG-SET: Distribute-Lists Example
+-----------------------------------------
|     ip prefix-list ROUTE permit 10.10.0.0/16
|     ip prefix-list SOURCE permit 10.5.1.1/32
|     distribute-list prefix ROUTE gateway SOURCE in         - Only accept 10.10.0.0/16 route from 10.5.1.1
|


CONFIG-SET: Extended Access-List Example (Prefix-List Equivalent)
+--------------------------------------------------------------------
|     access-list 100 permit ip host 10.5.1.1 host 10.0.0.0
|     distribute-list 100 in                                 - Only accept 10.0.0.0/8 route from 10.5.1.1
|


-----------
 COMMANDS
-----------
#router rip
 #offset-list [acl] {in|out} {offset} {interface}     - Increases the RIP metric by the offset

 #distribute-list {acl | prefix} {in|out}             - Filters all routes matching the ACL or prefix-list

 #distribute-list gateway {prefix-list} {in|out} {interface}
                                                      - Filters all routers to/from a neighbor

 #distribute-list prefix {prefix-routes} gateway {prefix-source} {in|out}
                                                      - Filters prefixes from a specific source from entering the routing table

 #distance {ad} {src-ip [mask]} [acl]                 - By setting the distance to 255, routes could be filtered
```

Default Routing

COMMANDS

```
#ip prefix-list DEFAULT seq 5 permit 0.0.0.0/0        - Creates a prefix-list matching a default route

#router rip
 #default-information originate                       - Generates and advertises an unconditional default route to neighbors
 #distribute-list prefix DEFAULT out [interface]      - Limit the advertisements sent out to only the default route
```

Authentication

- Only supported on RIPv2.
- Supports clear text and MD5.
- Configured using key-chains.
- RIP, unlike EIGRP, does not require the same key-number on both sides.
- When configuring, order of operation is important.
- When making changes to the key-chain, first remove the config of the interface.
- Steps involved:
  1- Define a key chain with a name.
  2- Define the key or keys on the key-chain.
  3- Enable authentication on an interface and specify the key-chain to be used.
  4- Specify whether the interfaces will use clear text or MD5. If not specified, clear is used.
  5- Optionally configure key management.

COMMANDS

```
# sh ip protocols | begin rip                         - Shows the key-chain in use

#key chain NAME                                       - Step 1: Defines a key-chain
  #key 1                                              - Step 2: Defines the key/s on the chain
  #key-string STRING                                  - Step 2: Specifies the key-string
#interface eth0
  #ip rip authentication key-chain NAME               - Step 3: Enable authentication on an interfaces by using the key-chain
  #ip rip authentication mode md5                     - Step 4: Specifies whether the interfaces will use clear text or MD5
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   Troubleshooting RIP
*=========================*
- When troubleshooting RIP updates and route-selection issues, consider the following:
 > Are the necessary RIP interfaces in an UP,UP state?                                   # sh ip int brief
 > Are all the interface IP addresses correct?                                           # sh int | i line|Internet
 > Is RIP version 2 enabled?                                                             # sh run | i version|rip
 > Is auto-summary disabled for RIP?                                                     # sh run | i no auto-summary
 > Are the correct network statements configured?                                        # sh run | i network
 > Is there layer2 connectivity and layer3 reachability?                                 # ping {neighbor-ip}
 > With frame-relay multipoint interfaces, is broadcast replication enabled?             # sh run | i frame.*map
 > Are you seeing a neighbor's routes in the RIP database?                               # sh ip rip database
 > Does another route with a lower AD from another protocol get installed in the RIB?    # sh ip route {prefix}
 > Is a RIP route flapping (look at Update Timer, is it always at 00:00:00)?             # sh ip route {prefix}
 > Is split-horizon enabled?                                                             # sh run | i interface|split
 > Is the use of "passive-interface" preventing route updates being sent out?            # sh run | i passive
 > Are any offset filters configured denying routes?                                     # sh run | i offset-list
 > Are any distribute lists configured denying the routes entry in the local RIB?        # sh run | i distribute-list
 > Is the distance command used to filter routes?                                        # sh run | i distance eigrp
 > Is summarization the cause of more specifics not being seen?                          # sh run | i summary-add
 > Do the routes redistributed into RIP have valid hop-counts defined?                   # sh run | i redist|default.*met
 > If authentication is configured, do the key-chain and key match?                      # sh run | s key-chain
 > As a last resort this debug is very handy to see what is going on.                    # debug ip rip
 ______   _   _______   ______    ______
|  ____) | | |  _____) (_____ \  (_____ \
| |___   | | | |  ___   _____) )  _____) )
|  ___)  | | | | (_  | |  __  /  |  ____/
| |_____ | | | |___) | | |  \ \  | |
|_______)|_|  \_____/  |_|   \_| |_|

*-=-=-=-=-=-=-=-=-=-=-=-=-*
|      INDEX              |
*-=-=-=-=-=-=-=-=-=-=-=-=-*
```

```
- EIGRP Operation
- Metrics, Timers and K-values
- Variance and Load-Sharing
      + EIGRP Unequal Load-Sharing
      + Multi-Interface Load Splitting
- Convergence Timers
- Routing Updates
- Packet Types
- DUAL Finite State
- Passive Interface
- Split-Horizon and Next-Hop-Self
- Authentication
      + MD5
      + Key Rotation
- Summarization
      + Auto-Summary
      + Interface
      + Default Routing
      + Floating Summary
- Filtering
      + Distribute-List
      + Offset List
      + Distance
- Stub Routing
- Bandwidth Percent
- Troubleshooting EIGRP
      + Router Configuration
      + Adjacencies
      + Route-Selection Issues
- Output-101
      + Overlapping Key-String
      + FD, AD, Metric
      + Stub Neighbor
      + "debug eigrp packet" Explained
      + EIGRP Neighbor Events
```

EIGRP Operation

- DOC-CD LOCATION
  - > 12.4T Configuration Guides
    - > Cisco IOS IP Routing: EIGRP Configuration Guide, Release 12.4T
      - > Configuring EIGRP

- Hybrid IGP using DUAL (Diffusing Update Algorithm).
- Uses own transport protocol: 88.
- Multicasts to destination 224.0.0.10 (TTL=0) using RTP, the receiving neighbor unicasts an acknowledgment.
- Unequal-cost load sharing up to 16 links.
- EIGRP does not form neighbors over secondary networks/IPs.

- Route entries are classified into one of three categories:
  - > Interior routes - A path to a subnet of the network address of the data link on which the update is being broadcast.
    - >> Interior route is 'local' to the major network to which the advertising and receiving router are commonly connected.
    - >> 192.168.2.192/26 is advertised to 192.168.2.64/26 within the same AS as an interior route because it falls within the same major network.
  - > System routes- A path to a network address that has been summarized by a network boundary router.
    - >> 192.168.3.0 is advertised to 192.168.2.0 within the same AS as a system route.
  - > Exterior routes- A path to a default network, or a network in another autonomous system.
    - >> 196.12.1.0 is advertised to 64.32.0.0 in a separate AS as an exterior route.

Metrics, Timers and K-values

- EIGRP calculates all metrics from outgoing interfaces only.
- The composite metric is the minimum bandwidth of an outgoing interface, the cumulative delay, load, reliability, and the smallest MTU along a path.
- Metrics:

```
 > BW        - Expressed in units of kilobits per sec.
             - Static number used for metric calculation only, doesn't reflect actual bandwidth.
             - To calculate EIGRP bandwidth metric amount: 10000000/configured BW.

 > DLY       - Static figure, expressed in units of microseconds.
             - To calculate EIGRP DLY metric amount: DLY/10.

 > REL       - Dynamically measured.
             - Is expressed as an 8-bit number, where 255=100% reliable link and 1=minimally reliable link.

 > LOAD      - Dynamically measured.
             - Is expressed as an 8-bit number, 1=minimally loaded link and 255=100% loaded link.
```

- Default K-Values: K1=1 K2=0 K3=1 K4=0 K5=0.
- EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + (K3*Delay)*(K5/(Reliability + K4))).

- By default, EIGRP chooses a route based ONLY on bandwidth and delay (due to the default k-values).
  > Default metric = 256 x [10^7 / (min(BW)) + (sum(DLY)) / 10].
- EIGRP supports hop-count merely as a way to prevent routing loops.

COMMANDS

```
#metric maximum-hops (number)               - Changes the default hop-count limit of 100. Values 1-225

#metric weights tos k1 k2 k3 k4 k5          - Changes the metric calculation of the K-values (K1=1 K2=0 K3=1 K4=0 K5=0)
#metric weights 0 0 0 1 0 0                 - Changes the metric calculation to only use DLY
#metric weights 0 1 0 0 0 0                 - Changes the metric calculation to only use BW

#interface eth0
 #bandwidth 64                              - Changes the bandwidth to 64 Kbit
 #delay 5                                   - Specifies delay in tens of microseconds, changes the delay to 50 usec
```

Variance and Load-Sharing

- The variance command is used to determine which routes are feasible for unequal-cost load sharing.
- Variance defines a multiplier by which a metric may differ, or vary, from the metric of the lowest-cost route.
- Any route whose metric exceeds the metric of the lowest-cost route, multiplied by the variance, will not be considered a feasible route.
- The default variance is one, meaning that the metrics of multiple routes must be equal in order to load balance.
- Variance can only be specified in whole numbers.
- Load sharing is per-destination if the packet is fast-switched or CEF-switched using the default CEF configuration.
- Load sharing is per-packet if process switching is used or if the CEF configuration was modified.
- CEF and fast-switching can be turned off with "no ip cef" and "no ip route-cache" and then the router will perform unequal-cost, per-packet load-balancing.

- EIGRP Unequal Load-Sharing
  > Is inversely based on the traffic-share rate amongst the multiple paths. I.e. with a traffic-share rate of 1:5, the first path would get five times more traffic than the second path.
  > This default behavior can be disabled with the command "no traffic-share balanced".
  > For a good EIGRP load-sharing example refer to: http://routing-bits.com/2009/04/02/eigrp-metric-manipulation/

- Multi-Interface Load Splitting
  > If multiple paths are available to the same destination, only paths with the minimum metric will be installed in the routing table.
  > This is a protocol independent command and not specific to EIGRP.
  > The number of static paths allowed are never more than six. The extra paths are ignored if more than six are available.

```
 > For dynamic routing protocols, the number of paths are controlled by the "maximum-paths" command.
 > Configured with the command "traffic-share min".
 > This command can be used instead of "variance 1", to see the available best paths in the routing table even though only
      the best paths should be used.
 > The 'across-interfaces' keyword optionally allows multiple interfaces to be used.

-----------
 COMMANDS
-----------
#no ip cef                                             - Disables CEF under the interface
#no ip route-cache                                     - Disables fast-switching under the interface
                                                           (both necessary for per packet load-balancing via process switching)
#router eigrp {asn}
 #variance {number}                                    - Allows a metric to be 5 times more than the current FD (def=1)
 #maximum-paths {number}                               - Sets the maximum number of parallel paths allowed in a routing table
 #traffic-share balanced                               - Distributes traffic proportionately to metric ratios (default = enabled)
 #traffic-share min across-interfaces                  - Choose only one best metric paths even though multiples exist


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Convergence Timers
*==========================*
- Never change the timers unless asked to.
- Hellos are sent using unicast every 60 sec on access links with speeds of T1 or slower.
- Hellos are sent using multicast every 5 sec on all other network links.
- The hold-time interval is 180 sec on low-speed NBMA networks.
- The hold-time interval is 15 sec on all other networks.

-----------
 COMMANDS
-----------
#sh ip eigrp neighbors                                 - Shows each neighbor in the neighbor table along wit the timers

#interface s0/0
 #ip hello-interval eigrp {ASN} {seconds}              - Changes the default hello interval
 #ip hold-time eigrp {ASN} {seconds}                   - Changes the default hold-time


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Routing Updates
*==========================*
- EIGRP updates are multicast to 224.0.0.10.
- EIGRP updates are non-periodic, partial (only changes) and bounded (only to relevant neighbors).

- Routing updates can be sent as unicast at a process level using the "neighbor" command.
- But both sides must be configured to use unicast.
```

```
!!NOTE!! If configured, EIGRP stops processing all multicast packets that are received on that interface.
       The router also stops sending EIGRP multicast packets on that interface.
!!NOTE!! Upon configuring, all sessions from that interface will be dropped.
```

- Using an ACL to filter EIGRP traffic between two neighbors is recommended.
- Packets sourced by a router are not passed through an outbound ACL by default.

COMMANDS

```
#router eigrp {asn}
 #neighbor {ip} {interface}                           - Defines a unicast session to a neighbor. Required on both sides

#ip access-list 100 deny eigrp any any                - Denies any EIGRP traffic
#ip access-list 100 permit ip any any                 - Permits all other traffic
#interface eth0
 #ip access-group 100 in                              - Applied inbound, as outbound would have no effect
```

Packet Types

- EIGRP uses multiple packet types, they are all identified by protocol number 88 in the IP header.

```
 > Hellos                  - Are used by the neighbor discovery and recovery process. Hellos are unicast or multicast and
                               uses unreliable delivery.
 > ACKs                    - Are hello packets with no data in them. ACKs are always unicast and use unreliable delivery.
 > Updates                 - Convey route information. Updates could be unicast/multicast and always use reliable delivery.
 > Queries/Replies         - Used by DUAL for computations. Queries can be unicast or multicast, but replies are always unicast.
```

- Any reliable multicast packets sent that were not acknowledged by the neighbor they were sent too, will be followed by a retransmitted unicast packet to that neighbor.
- If an acknowledgement was not received after 16 of these unicast retransmissions, the neighbor will be declared dead.
- Retransmission TimeOut (RTO) is the time between the subsequent unicasts.
- Smooth Round-Trip Time (SRTT) is the time, between a packet sent to the neighbor and the receipt of an acknowledgment.

DUAL Finite State

- The lowest calculated metric to each destination will become the FD (Feasible Distance) of that destination.
- The FC (Feasibility Condition) is a condition that is met if a neighbor's AD (Advertised Distance) to a destination is lower than the router's current FD to that same destination.
- If a neighbor's AD to a destination meets the FC, that neighbor becomes a FS (Feasible Successor) for that destination.
- Because FSs are always 'downstream' a router will never choose a path that will lead back through itself, thus avoiding creating a loop.
- Such a loop path would have a distance larger than the FD.

- Every destination for which one or more FS exist, will be recorded in a topology table.
- After insertion each route, when no diffusing is taking place, will be in a passive state.
- If there are two successors with a locally-calculated metric equal to the FD, both routes are entered into the route table, and equal-cost load balancing will be performed.

- If a link to a successor fails(input event), or if the cost of the link increases beyond the FD (input event), the router will first look into its topology table for an FS.
- If an FS is found, through local computation, it will become the successor. This occurs in the sub-second range. An update is sent to all neighbors and the route remains in the passive state.
- If an FS cannot be found in the topology table, the router will begin a diffusing computation by querying neighbors for possible routes and the route will change to the active state.
- For each neighbor to whom a query is sent, the router will set a reply status flag (r) to keep track of all outstanding queries.
- The diffusing computation is complete when the router has received a reply to every query sent to every neighbor.
- If all expected replies are not received before the Active time expires, the route is declared SIA (Stuck-In-Active).
- At the completion of the diffusing computation, the originating router will set the FD to infinity to ensure that any neighbor replying with a finite distance to the destination will meet the FC and become an FS.
- Remember that queries cause the diffusing calculation to grow larger, whereas replies cause it to diminish/grow smaller.

COMMANDS

```
# sh ip eigrp topology                                - ONLY route via 10.1.3.1 is in the route table since it has the lowest FD
   P 10.1.2.0/24, 2 successors, FD is 768             - The lowest metric to subnet 10.1.2.0 is 768, so 768 is the FD
      via 10.1.3.1 (768/256), Serial0                 - The first number is the locally calculated metric to the destination
      via 10.1.5.2 (1280/512), Serial1                - The second number is the metric advertised by the neighbor (AD)

# sh logging | i SIA                                  - The logging buffer would show when a route is SIA
 %DUAL-3-SIA: Route 10.1.1.0/24 stuck-in-active state in IP-EIGRP 1. Cleaning up

#timers active-time {minutes | disabled}              - Changes the default (180 sec) SIA timer
```

-=-

============================

Passive Interface

============================

- The "passive-interface" command prevents EIGRP hellos from being sent on data links where they don't belong.
- Will prevent neighbor establishments and routes being advertised, as received hellos will be ignored.

COMMANDS

```
#router eigrp {asn}
 #passive-interface default                           - Disables all interfaces from sending hellos unless explicitly allowed
 #passive-interface {int}                             - Disables one interface from sending hellos
 #no passive-interface {int}                          - Enabled sending hello from the specified interface
```

Split-Horizon and Next-Hop-Self

- Is a method of preventing routing loops by prohibiting a router from advertising a route back onto the interfaces from which the route was learned.
- It is always enabled with EIGRP.
- It might be necessary to diable split-horizon with multi-point frame-relay interfaces.

- Next-Hop-Self
  > EIGRP will, by default, set the IP next-hop value to be itself for routes that it is advertising, even when advertising those routes back out the same interface where it learned them.
  > This default behavior can be disabled by instructing EIGRP to use the received next hop value when advertising a routes.
  > This might be necessary in some unique spoke-to-spoke scenarios.
  > Configured with "no ip next-hop-self eigrp"

COMMANDS

```
#interface s0/0
 #no ip split-horizon eigrp {ASN}                      - Disables split-horizon
```

Authentication

- EIGRP packets can ONLY be authenticated using an MD5 cryptographic checksum.
- Configured using key-chains.
- EIGRP, unlike RIP, requires the same key-number on both sides.
- When configuring, the order of operation is important.
- When doing changes to the key-chain, first remove the key-chain off the interface.
- The steps for configuring EIGRP authentication are:
 1- Define a key-chain with a name.
 2- Define the key or keys on the key-chain.
 3- Enable authentication on an interface and specify the key-chain to be used.
 4- Optionally configure key management.

COMMANDS

```
# sh key chain {name}                                  - Shows the configured keys and which are currently valid
# debug eigrp packet hello                             - Shows received authentication packets.

#key chain {name}
 #key {key number}
  #key-string {string}
  #send-lifetime {from H:M:S MON DAY YEAR} {to H:M:S MON DAY YEAR}      - Specifies the period a key is valid for
  #accept-lifetime {from H:M:S MON DAY YEAR} {to H:M:S MON DAY YEAR}    - Specifies overlapping times for a key to be accepted
```

```
#interface s0/0
 #ip authentication key-chain eigrp {ASN} {chain name}                  - Assigns the key-chain to the interface
 #ip authentication mode eigrp {ASN] md5                                - Specifies MD5


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Summarization
*==========================*
- EIGRP, by default, auto-summarizes prefixes to the classful boundary when passing major a network boundary, but this can be
      disabled. If auto-summary is enabled, interfaces are summarized at the class boundary.
- A route to Null0 for summary routes is created to prevent black-holes.
- Disabling automatic summarization can prevent ambiguous routing between similar network subnets and is always recommended.

- Manual summarization for EIGRP is interface-specific.
 > This provides the flexibility to be able to advertise different summary routes out of different interfaces for the same process.
- Manual summarization is configured with the "ip summary-address eigrp" command.
 > By default this will automatically suppress the advertisement of the more specific networks and create a route to Null0
 > To have more specific routes sent, use a leak-map.
- The summary routes advertised into EIGRP are not tagged as external routes, like OSPF.
- The floating summary route is created by applying a default route and an administrative distance at the interface level.

-----------
 COMMANDS
-----------
#no auto-summary                                          - Disables auto-summary to the classful boundary when passing
                                                                     between major network boundaries (default = Enabled)
#interface eth0
 #ip summary-address eigrp {ASN} {aggregate} [leak-map] [AD]
                                                          - Automatically suppresses the advertisement of the more specific networks
                                                          - Specifies the summary, mask and the process into which the
                                                               summary is to be advertised
                                                          - [leak-map]:Route-map allows more specific routes and the summary
                                                               to be advertised

 #ip summary-address eigrp 100 0.0.0.0 0.0.0.0 250        - Example of a floating summary route with a higher AD.


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Filtering
*==========================*
-----------
 COMMANDS
-----------
#offset-list [ACL] {in|out} {offset} {interface}          - Increases the EIGRP composite metric value

#distribute-list {ACL |  refix} {in|out}                  - Filters all routes matching the ACL or prefix-list
```

```
#distribute-list gateway {prefix-list} {in|out} {interface}
                                                      - Filters all routes to/from a neighbor

#distribute-list prefix {prefix-routes} gateway {prefix-source} {in|out}
                                                      - Filters prefix from a specific source from entering the routing table

#distance eigrp {ad-internal} {ad-external}           - Changes the distance for both internal and external EIGRP routes


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Stub Routing
*==========================*
- A router that has EIGRP stub neighbors will not send queries to those stubs, thereby eliminating the chance that a stub
    will cause SIA (stuck-in-active) conditions and routing instabilities in other parts of the network.
- Stub routing can also be useful in preventing a router from being used as transit/backup by only sending local updates
    not containing remote-learned routes.

-----------
 COMMANDS
-----------
# sh ip eigrp neighbors [detail]                      - [detail] 'CONNECTED SUMMARY' shows the configured STUB neighbors

#eigrp stub [connected | redistributed | static | summary | receive-only]
                                                      - Configured on a stub router defining which routes to be sent
                                                      - [receive-only] The stub router won't send any route information in updates
                                                      - Default= Only updates containing connected and summary routes will be sent


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Bandwidth Percent
*==========================*
- EIGRP is designed to use no more than 50% of the available bandwidth of a link.
- This restriction means that EIGRP's pacing is tied to the configured bandwidth.

- Example:
 > Suppose an interface is connected to a 512K serial link, but the bandwidth is configured at 128K.
 > By default EIGRP would limit itself to 50% of the configured amount,  in this case 64K.
 > The command below adjusts the EIGRP bandwidth percent to 200% of 128K, which is 256K, half of the actual link bandwidth.

-----------
 COMMANDS
-----------
#interface s0/0                                 - Assumes the physical clock is 512k
 #bandwidth 128
 #ip bandwidth-percent eigrp 1 200              - Adjusts the EIGRP bandwidth percent to 200% of 128K
                                                - That is 256K, half of the actual link bandwidth 512k
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Troubleshooting EIGRP
*==========================*
- When examining an individual router's configuration, consider the following:
 > Are the necessary EIGRP interfaces in an UP,UP state?                              # sh ip int brief
 > Are all the interface IP addresses and masks correct?                              # sh int | i line|Internet
 > Are the correct EIGRP autonomous system numbers configured?                        # sh run | i router eigrp
 > Are the correct network statements configured?                                     # sh run | i network
 > Is auto-summary disabled for EIGRP?                                                # sh run | i no auto-summary
 > Is every router using the correct router-id? Any duplicates?                       # sh run | i eigrp router-id


- When examining adjacencies (or the lack thereof), consider the following:
 > It could be helpful to log the neighbor adjacency changes.                         #eigrp log-adjacency-changes
 > Is the router attempting to form an adjacency with another's secondary address?    # sh run | i network
 > Is there layer2 connectivity and layer3 reachability?                              # ping {neighbor-ip}
 > With frame-relay multipoint interfaces, is broadcast replication enabled?          # sh run | i frame.*map
 > Are any access-lists dropping protocol-88 traffic or any neighbor specific IPs.    # sh ip interface | i line|list
 > Are hellos being sent from both neighbors and received by both?                    # debug eigrp packets hello
 > Do the K-values match between the neighbors (thit is required)?                    # sh ip prot | i weight
 > Does the EIGRP autonomous system numbers configured match between neighbors?       # sh run | i router eigrp
 > Is only one side of a link configured to unicast updates?                          # sh run | i neighbor
 > Is the use of "passive-interface" preventing a neighbor adjacency?                 # sh run | i passive
 > If authentication is configured, do the key-chain and key match?                   # sh run | s key-chain
 > Was the key-chain applied to the interface?                                        # sh ip eigrp int detail | i Auth
 > If lifetime was specified, is the key-chain still active (look for 'valid now')?   # sh key chain {name}
 > Examine the counters from the EIGRP neighbor list:                                 # sh ip eigrp neighbors
      >> SRTT- A value of 0 indicates that a packet has never made the round trip.    - Reachability
      >> Q Count- Are there packets queued for transmission (Q should be = 0)?        - Link issues
      >> Seq Num- A value of 0 indicates that no reliable packets have ever been received.- Reachability or filtering


- When troubleshooting route-selection issues, consider the following:
 > A handy command to see routes inserted and pulled from the RIB is:                 # debug ip routing
 > Are the expected routes appearing in the EIGRP topology table?                     # sh ip eigrp topology
 > Are any offset filters configured denying routes?                                  # sh run | i offset-list
 > Are any distribute lists configured denying the routes entry into the local RIB?   # sh run | i distribute-list
 > Is the distance command used to filter routes?                                     # sh run | i distance eigrp
 > For a EIGRP route that is not installed, was the FC (Feasible Condition) met?            # sh ip eigrp topology
 > Does another route with a lower AD from another protocol get installed in the RIB?  # sh ip route {prefix}
 > Is an EIGRP route flapping (look at Update Timer, is it always at 00:00:00)?       # sh ip route {prefix}
 > Is a neighbor in the forwarding path configured as a stub? (look for CONNECTED SUMMARY)?     # sh ip eigrp neighbors detail
 > Do routes redistributed into EIGRP have the composite metrics defined?             # sh run | i redist|default.*met
 > Is summarization the cause of more specifics not being seen?                       # sh run | i summary-add
 > In a hub-and-spoke design, does the hub-interface have EIGRP split horizon disabled?   # sh run | i eigrp.*split
 > Flapping neighbors and intermittent reachability could point to SIA routes.              # sh log | i SIA
 > Common causes of SIAs in larger EIGRP networks are:
      >> Heavily congested links and/or low-bandwidth data links.
      >> Routers with low memory or over-utilized CPUs.
      >> Careless adjustment of the bandwidth parameter on an interface.


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
    OUTPUT-101
*============================*
--------------------------------------
      # sh key chain EIGRP
--------------------------------------
key-chain EIGRP:
    key 1 -- text "cisco123"
        accept lifetime (00:00:00 UTC Jan 1 2000) - (14:25:00 UTC Sep 20 2008) [valid now]
        send lifetime (00:00:00 UTC Jan 1 2000) - (14:10:00 UTC Sep 20 2008) [valid now]
    key 2 -- text "cisco456"
        accept lifetime (14:05:00 UTC Sep 20 2008) - (infinite) [valid now]   - Overlapping key-string allowed 5 minutes earlier
        send lifetime (14:10:00 UTC Sep 20 2008) - (infinite)


---------------------------------------------------
      # sh ip eigrp interfaces detail
---------------------------------------------------
IP-EIGRP interfaces for process 100
                        Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Se1/0              1        0/0        83       2/95          439           0
  Hello interval is 60 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 12/22
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 5
  Retransmissions sent: 4  Out-of-sequence rcvd: 1
  Authentication mode is md5,  key-chain is "EIGRP"                    - Shows key-chain used with neighbor


-----------------------------------------------------
      # sh ip eigrp topology all-links
-----------------------------------------------------
IP-EIGRP Topology Table for AS(100)/ID(155.1.4.4)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 155.1.5.0/24, 1 successors, FD is 10639872, serno 20      - Lowest metric to subnet is 10639872, so this is the FD
        via 155.1.0.5 (10639872/128256), Serial1/0          - First number in the () is the locally metric to the destination
        via 155.1.45.5 (40640000/128256), Serial1/1         - Second number is the metric advertised by the neighbor (AD)
P 155.1.45.0/24, 1 successors, FD is 40512000, serno 25
        via Connected, Serial1/1
        via 155.1.0.5 (41024000/40512000), Serial1/0
```

```
---------------------------------------------------
      # sh ip eigrp neighbors detail
---------------------------------------------------
IP-EIGRP neighbors for process 100

H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   155.1.45.4              Se1/1             11 00:00:16   35  2280  0  77
   Version 12.3/1.2, Retrans: 0, Retries: 0, Prefixes: 2

   Suppressing queries
1   155.1.0.4               Se1/0            148 00:02:21   17   570  0  73
   Version 12.3/1.2, Retrans: 23, Retries: 0, Prefixes: 2
   Stub Peer Advertising ( CONNECTED SUMMARY ) Routes                   - Shows neighbor is configured as a Stub
   Suppressing queries
```

```
----------------------------------------------------------
      # debug eigrp packet update query reply
----------------------------------------------------------
    EIGRP Packets debugging is on (UPDATE, QUERY, REPLY)
    %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, changed state to down
      ....
      EIGRP: Enqueueing QUERY on Serial0 iidbQ un/rely 0/1 serno 45-49
      EIGRP: Enqueueing QUERY on Serial0 nbr 10.1.6.1 iidbQ un/rely 0/0 peerQ un/rely 0/0 serno 45-49
      EIGRP: Sending QUERY on Serial0 nbr 10.1.6.1
        AS 1, Flags 0x0, Seq 45/64 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno 45-49
      EIGRP: Received REPLY on Serial0 nbr 10.1.6.1
        AS 1, Flags 0x0, Seq 65/45 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/0
      ....
      EIGRP: Received HELLO on Ethernet0 nbr 192.168.1.1
         AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
      EIGRP: Sending UPDATE on Ethernet0 nbr 192.168.1.1, retry 15, RTO 5000
         AS 75, Flags 0x1, Seq 22/0 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno

      EIGRP: Received HELLO on Ethernet0 nbr 192.168.1.1
         AS 75, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
      EIGRP: Sending UPDATE on Ethernet0 nbr 192.168.1.1, retry 16, RTO 5000
         AS 75, Flags 0x1, Seq 22/0 idbQ 1/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno
```

- Flags, in the debug messages, indicate the state of the flags in the EIGRP packet header:
 > 0x0 indicates that no flags are set.
 > 0x1 indicates that the initialization bit is set. This flag is set when the enclosed route entries are the first in a new neighbor relationship.
 > 0x2 indicates that the conditional receive bit is set. This flag is used in the proprietary reliable multicasting algorithm
 > Copy descretly owned by Kane Bagwell

- Other Flags:

```
 > Seq      - Is the Packet Sequence Number/Acknowledged Sequence Number.
 > idbq     - Indicates packets in the input queue/packets in the output queue of the interface.
 > iidbq    - Indicates unreliable multicast packets awaiting transmission/reliable multicast packets awaiting transmission
                on the interface.
 > peerQ    - Indicates unreliable unicast packets awaiting transmission/reliable unicast packets awaiting transmission on the
                interface.
 > serno    - Is a pointer to a doubly linked serial number for the route. This is used by proprietary mechanism for tracking
                the correct route information.
 > Retry    - Shows the Retransmission retry number, amount of re-attempts to send updates, without acknowledgements
 > RTO      - Shows the Retransmission Time-Out
```

```
------------------------------------------------------
      # debug eigrp neighbors 75 192.168.1.1
------------------------------------------------------
   IP Neighbor target enabled on AS 75 for 192.168.16
   IP-EIGRP Neighbor Target Events debugging is on
      EIGRP: Retransmission retry limit exceeded
      EIGRP: Holdtime expired
      EIGRP: Neighbor 192.168.1.1 went down on Ethernet0
      EIGRP: New peer 192.168.1.1
      EIGRP: Retransmission retry limit exceeded
      EIGRP: Holdtime expired
      EIGRP: Neighbor 192.168.1.1 went down on Ethernet0
      EIGRP: New peer 192.168.1.1
```

- Shows EIGRP neighbor events
- Optionally, the AS-Number and the neighbor IP could be specified to filter the output.

THIS PAGE WAS LEFT BLANK INTENTIONALLY

```
  _____      ______    _____     _______
 / ___ \    / _____)  (_____ \  (_______)
| |   | |  ( (____     _____) )  _____
| |   | |   \____ \   |  ____/  |  ___)
| |___| |   _____) )  | |       | |
 \_____/   (______/   |_|       |_|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-*
|       INDEX           |
*-=-=-=-=-=-=-=-=-=-=-=-*
```

- OSPF Overview
- Hello Protocol
    - Fast-Hello
- Advertising Routes
- Network Types
    - Broadcast
    - Non-Broadcast
    - Point-to-Multipoint
    - Point-to-Multipoint Non-Broadcast
    - Point-to-Point
    - Mismatch
- DR and BDR
- OSPF Finite State Machine
- Router Types
- LSA (Link State Advertisement)
    - LSA Types
    - LSA Overload Protection
    - LSA Throttling
    - LSA Timers and Pacing
- Area Types
    - Stub
    - Totally Stub
    - NSSA
    - Totally NSSA
    - FA Suppression in Translated Type-5 LSAs
- Filtering
    - Filter-List
    - Distribute-List
    - Distance
    - With Summarization
- Summarization
    - Inter-Area
    - External
- Stub Router Advertisement
- Passive-Interface
- Originating a Default Route
    - Unconditional Default Route
    - Conditional Default Route
    - Conditional Default Route with a Route-Map
    - OSPF Stub Areas

```
      + OSPF NSSA Default Routes
- Path Selection
      + Auto-Cost
      + Cost
      + Bandwidth
      + Neighbor Cost
      + iSPF (OSPF Incremental SPF)
- Authentication
      + Area
      + Interface
      + Virtual-Link
- OSPF Demand Circuit
- Troubleshooting OSPF
- OUTPUT-101
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
   OSPF Overview
*========================*
```

- DOC-CD LOCATION
     > 12.4T Configuration Guides
      > Cisco IOS IP Routing: OSPF Configuration Guide, Release 12.4T
       > Configuring OSPF

- Uses own transport protocol: 89.
- OSPF supports equal-cost load balancing for more efficient use. This is not an OSPF limitation instead it is set by the vendor depending on their hardware platform. For most IOS versions this limit is either six or eight paths.
- OSPF supports the use of route tagging for the tracking of external routes.
- OSPF packets are exchanged only between neighbors on a network. They are never routed beyond the network on which they originated.
- OSPF multicast packets use a TTL of 1.

- OSPF sees secondary networks as stub networks. This means no adjacencies will be established using secondary networks.
- If the network statement matches the IP address of the primary interface, the primary interface and the IP-unnumbered interface will have OSPF enabled.

```
-----------
 COMMANDS
-----------
# sh protocols                                 - Shows the interface prefixes and status
# sh ip protocols                              - Shows information about each enabled protocol
# sh ip ospf                                   - Shows information about OSPF timers, areas, etc
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Hello Protocol
*=====================*
- The hello protocol serves several purposes
 > It is a means by which neighbors are discovered.
 > It advertises several parameters on which two routers must agree before they can become neighbors.
 > Hello packets also act as keepalives between neighbors.
 > It ensures bidirectional communication between neighbors once a neighbor sees its own router ID in a received hello.
 > It elects DRs (Designated Routers) and BDRs (Backup Designated Routers) on Broadcast and NBMA networks.

- Each hello packet contains the following information.
 > Router ID of the originating router.
 > Area ID of the originating router interface.
 > Address mask of the originating interface.
 > Authentication type and information of the originating interface.
 > Hello-interval of the originating interface.
 > Router dead-interval of the originating interface.
 > Router priority.
 > DR and BDR.
 > Five flag bits signifying optional capabilities.
 > Router IDs of the originating router's neighbors.


- To establish adjacencies the following values must match the values configured on the receiving interface
 > Area ID.
 > Authentication.
 > Network mask (point-to-point links are the exception).
 > Hello-interval and Dead-interval.
 > MTU.
 > Options.

- Hello-Interval
 > OSPF-speaking routers periodically send a hello packet out of each OSPF-enabled interface.
 > Uses a default hello-interval of 10 seconds for broadcast and 30 sec for non-broadcast networks.
 > Configured on a per interface basis with "ip ospf hello-interval".

- Router Dead-Interval
 > Is the period of time after which a router will declare a neighbor down, if it does not receive a hello from that neighbor.
 > The default is four times the hello-interval but can be changed with the command "ip ospf dead-interval" below.

- By changing the hello manually with "ip ospf hello-int", the dead-interval is adjusted accordingly to 4x the new hello value.

- Fast-Hello Packets
 > Provides a way to configure the sending of hello packets in intervals less than 1 sec.
 > This is achieved using the "ip ospf dead-interval minimal" command by setting the dead interval to 1 sec.
 > The 'hello-multiplier' value is set to the number of hello packets that must be sent during that 1 sec.
 > Example:
      #ip ospf dead-interval min hello-multiplier 5         - Five hellos are sent every second i.e. at an interval of 200ms.
```

```
-----------
 COMMANDS
-----------
# sh ip ospf neighbor                           - Shows information from the neighbor data structure
                                                - Shows all OSPF speaking neighbors, their state, dead-timer, connected interfaces
# sh ip ospf interface                          - Shows OSPF-related interface information, DR, BDR, etc
# sh ip ospf interface brief                    - Shows brief summary of which interface is running which OSPF areas

#interface s0/0
 #ip ospf hello-interval {1-65535 sec}          - Specifies how often hellos are sent (10 sec/broadcast and 30 sec/non-broadcast)
 #ip ospf dead-interval {1-65535 sec | minimal} - How long to wait before declaring a neighbor dead (default = 4x hello-interval)
 #ip ospf dead-interval min hello-multiplier {no} - Configures OSPF fast hello
 #ip ospf mtu-ignore                            - Disables the MTU check. Used when a switch uses a different system MTU
                                                - The MTU size in a hello must be the same on between neighbors



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================*
   Advertising Routes
*=======================*
- There are three ways to advertise using OSPF:
 > Network area process command.
 > Interface command.
 > Redistribution.

- Network Area Command
 > Serves two purposes:
      >> Defines the interfaces on which OSPF runs.
      >> Defines the area membership of the interface.
 > Configured with "network {ip-address} {wildcard} area {area-id}" under the OSPF process.
 > The IP address and wildcard arguments together allow you to define one or multiple interfaces to be associated with a specific
      OSPF area using a single command.
 > The matched interfaces' IP-address/subnet-mask is advertised by OSPF, NOT the IP address/wildcard-mask of the "network" command.

- Interface Command
 > Accomplishes the same as the network area command.
 > Switches do not support this command.
 > Configured with "ip ospf {pid} area {area-id}" under the interface.


CONFIG-SET: Enabling interfaces to run OSPF
+---------------------------------------------------
|     interface s0
|      ip address 10.10.1.1 255.255.255.252
|      ip ospf 1 area 3                                 - Enables OSPF on serial 0 in area-3
|     !
|     interface s1
|      ip address 10.10.1.5 255.255.255.252
|     !
|     interface s2
|      ip address 10.5.2.2 255.255.255.192
|      !
|     interface s3
|      ip address 10.5.2.130 255.255.255.192
|      !
|     router ospf 1
|      network 10.10.1.5 0.0.0.0 area 1                        - Matches interface 1 only
|                                                              - Enables OSPF on serial 1 in area-1
|      network 10.5.2.0 0.0.0.255 area 2                       - Matches interface 2 and 3
|                                                              - Enables OSPF on serial 2/3 in area-2
|
```

```
-----------
 COMMANDS
-----------
# sh ip ospf interface [brief]                 - Shows OSPF enabled interfaces, DR, BDR, timers, type, etc

#router ospf {pid}
 #network {ip} {wildcard} area {area-id}       - Network command syntax that enables OSPF on an interface
 #network 0.0.0.0 0.0.0.0 area {area-id}       - Will enable OSPF on- and advertise any interface in an UP state

#interface s0
 #ip ospf {pid} area {area-id} [second none]   - Interface command syntax that enables OSPF on an interface
                                               - [secondaries none] Prevents advertises secondary IP addresses
```

-=-

=====================

Network Types

=====================

- An OSPF router maintains a data structure for each OSPF-enabled interface.
- If the network type is changed, the hello and dead timers will be adjusted accordingly.

- OSPF defines six network types:

 > Broadcast Network
      >> The default network type on ethernet interfaces.
      >> Will elect a DR and a BDR.
      >> Uses the multicast MAC 224.0.0.5 (0100.5E00.0005) for AllSPFRouters and 224.0.0.6 (0100.5E00.0006) for AllDRouters.
      >> There is NO next-hop modification. The next-hop IP remains that of the originating router.
      >> Layer3 to layer2 resolution is required.
      >> Broadcast networks can't have unicast neighbors configured.
      >> 10 hello / 40 dead-interval.

 > Non-Broadcast Network
      >> Can connect more than two routers but haS no native broadcast capability.
      >> Non-Broadcast is the default network type on multipoint frame-relay interfaces, e.g. a main interface.
      >> OSPF routers on NBMA networks elect a DR and BDR, but all OSPF packets are unicast between each manually specified neighbor with the "neighbor" command.
      >> The next-hop IP is not changed and remains the IP address of the originating router.

```
    >> The default priority is 1, and should be disabled (=0) on ALL SPOKES, to prevent a spoke from becoming a blackhole DR/BDR.
    >> 30 hello / 120 dead-interval.

> Point-to-point Network
    >> Default on T1, DS-3, SONET links and on point-to-point sub-interfaces on frame-relay.
    >> Has no DR/BDR election, OSPF configured is as per normal.
    >> Uses the multicast destination to AllSPFRouters (224.0.0.5), except for retransmitted LSAs, which are unicast.
    >> The next-hop IP is that of the advertising router.
    >> OSPF ignores subnet mask mismatch on point-to-point links.
    >> 10 hello / 40 dead-interval.

> Point-to-multipoint Network
    >> Cisco proprietary and not a default option but arguably the best choice for NBMA networks.
    >> Special configuration of NBMA networks in which the networks are treated as a collection of point-to-point links.
    >> Does not elect a DR and BDR, and the OSPF packets are multicast (224.0.0.5) to each known neighbor.
    >> The next-hop IP is that of the advertising neighbor.
    >> Layer3 to layer2 resolution is ONLY needed for the directly-connected neighbors.
    >> Non-direct neighbors use recursive layer3 IP routing to reach each other.
    >> In addition, the endpoints of point-to-multipoint networks are advertised as host routes (/32).
    >> 30 hello / 120 dead-interval.

> Point-to-Multipoint Non-Broadcast Network
    >> Is Cisco proprietary. It is the same as point-to-multipoint, but configured with the additional 'non-broadcast' keyword.
    >> No DR/BDR election, uses unicast appose to multicast, to each manually specified neighbor.
    >> As a result, the directly connected neighbor must be manually defined with the "neighbor" command. It is only
         required on one side, but it is best to do it on both sides.
    >> The next-hop IP is that of the advertising neighbor.
    >> IP routing will be used to establish reachability between devices that are non-adjacent at layer2.
    >> This network was created to allow for the assignment of the cost per neighbor instead of using the interface's cost.
    >> Remember that the cost is based on the 'incoming' interface's bandwidth and not the bandwidth of the
         neighbor's interface.
    >> 30 hello / 120 dead-interval.

> Virtual Links
    >> Used to link an area to the backbone through a non-backbone area (also known as a transit area).
    >> Can also be used to connect two parts of a partitioned backbone through a non-backbone area.
    >> Must be configured between two ABRs, of which one must be connected to area 0.
    >> The transit area may not be a stub area and must have full routing information.
    >> The virtual link will transition to the fully functional point-to-point interface state when a route to
          the neighboring ABR is found in the route table.
    >> OSPF ignores subnet mask mismatch on point-to-point links.
    >> A virtual link is seen as an interface in area 0.
    >> All area 0 attributes are inherited by routers attached to the virtual link, including summarization and authentication.
    >> The cost of the virtual link is the cost of the route to the neighbors interface via the transit area.
    >> To see the cost of using the transit area use "sh ip ospf virtual-link" and refer to 'cost of using'.

> OSPF over GRE
    >> OSPF virtual links may not transit stub areas.
    >> If a virtual link over a stub area is required, the only solution is to use a GRE tunnel.
    >> The tunnel interface must have an IP address with a matching network statement in area 0.
```

```
 > Stub/Loopback Network
      >> Default for loopback interfaces.
      >> Assumes only a single attached router. OSPF advertises stub networks as host routes(/32).
      >> Don't confuse this with stub areas!

-----------
 COMMANDS
-----------
# sh ip ospf neighbors                            - Shows the OSPF neighbors, state, interface, etc
# sh ip ospf interface [brief]                    - Shows OSPF related interface information, DR, BDR, timers, type, etc
# sh ip ospf virtual-link                         - Shows the state of a virtual link, the cost of transit area, transit interface

#interface s0
 #ip ospf {pid} area {area-id}                    - Same as OSPF network command. Places the interface in a specified area
 #ip ospf network broadcast                       - Changes the network type to broadcast. Timers: 10/40
 #ip ospf network non-broadcast                   - Changes the network type to NBMA. Timers: 30/120. Require manual neighbors
 #ip ospf network point-to-point                  - Changes the network type to point-to-point. Timers: 10/40
 #ip ospf network point-to-multipoint             - Changes the network type to point-to-multipoint. Timers: 30/120
 #ip ospf network point-to-multi [non-broadcast]- Changes to network type to point-to-multipoint non-broadcast. Timers: 30/120
 #ip ospf priority {number}                       - Highest priority wins (Default = 1, Ineligible = 0)

#router ospf 1
 #network {ip} {mask} area {area-id}              - Defines an interface on which OSPF runs and its area ID
 #area {transit-area} virtual-link {ABR-ID}       - Configures one end of the virtual link. {ABR-RID} = Area Border Router ID
 #neighbor {ip} [priority {pri}] [cost {cost}]    - Manually specifies a neighbor
                                                  - Optionally define priority or cost for the neighbor


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   DR and BDR
*=====================*
- Will be elected on broadcast and NMBA networks.
- Addressing
 > All DROther routers send updates to the destination multicast address AllDRouters (224.0.0.6) (0100.5E00.0006).
 > All DR/BDR routers send updates to the destination multicast address AllSPFRouters (224.0.0.5) (0100.5E00.0005).
- The concept behind the DR is that the broadcast link itself is considered a 'pseudonode'.
- The cost from an attached router to the pseudonode is the outgoing cost of that router's interface to the broadcast link.
- The cost from the pseudonode to any attached router is zero.
- The DR is a property of a router's interface, not the entire router.
- On broadcast segments traffic doesn't flow through the DR, only updates are sent to the DR and BDR.
- The DR/BDR must have layer2 connectivity to all neighbors.

- Router Interface Priority:
 > Influences the election process between DR and BDR, but will not override an active DR or BDR.
 > OSPF elections do not support pre-emption.
 > Highest priority value wins. The default priority is 1.
 > Routers with a priority of 0 are ineligible to become the DR or BDR.
 > The priority can be changed on a per-multi-access-interface basis with the command "ip ospf priority".
```

```
- Router ID
 > Could be used as a tie-breaker when router priorities are equal.
 > Is the highest loopback IP in an 'UP' state. If no loopbacks are configured, it is the highest interface IP in an 'UP' state.
 > Can be statically set.

-----------
 COMMANDS
-----------
#interface e0
 #ip ospf priority {priority}                     - Highest router priority wins the DR/BDR election. (Default=1, Ineligible=0)
 #router-id {id}                                  - Manually assign an OSPF router ID, to be configure before any other OSPF config


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
   OSPF Finite State Machine
*==============================*
- An OSPF router transitions a neighbor through several states before the neighbor is considered fully adjacent.
 > Down
      >> The initial state of a neighbor conversation indicates that no hellos have been heard from
            the neighbor in the last router dead-interval.
      >> If a neighbor transitions to the DOWN state, the link state retransmission, database summary,
             and link state request lists are cleared.

 > Attempt
      >> This state applies only to neighbors on NBMA networks, where neighbors are manually configured.
      >> A router sends packets to a neighbor in Attempt state at the hello-interval instead of the poll-interval.

 > Init
      >> This state indicates that a hello packet has been seen from the neighbor in the last router dead-interval,
          but two-way communication has not yet been established.

 > 2Way
      >> Indicates that the router has seen its own router ID in the neighbor field of the neighbor's hello packets,
            meaning bidirectional conversation has been established.
      >> On multi-access networks, neighbors must be in this state or higher to be eligible to be elected as the DR or BDR.

 > ExStart
      >> The router and its neighbor will establish a master/slave relationship and determine the initial DD sequence number
            to exchange Data Descriptor Packets (DDPs).
      >> The neighbor with the highest router ID becomes the master.

 > Exchange
      >> The router sends DDPs describing in summary its entire link state database to neighbors that are in the Exchange state.
      >> The router may also send link state request packets, requesting more recent LSAs to neighbors in this state.

 > Loading
      >> The router sends link state Request packets to neighbors, requesting more recent LSAs that have been discovered
            in the exchange state but have not yet been received.
```

- > Full
  - >> Neighbors in this state are fully adjacent, and the adjacencies appear in router LSAs and network LSAs.

- The adjacency building process uses four OSPF packet types
  - > DDP – Database Description Packets (type 2)
    - >> Carry a summary description of each LSA in the originating router's link state database. These descriptions are not the complete LSAs.
    - >> Three flags in the DDP are used to manage the adjacency building process:
      - >>> I-bit, or Initial bit, when set indicates the first DDP sent.
      - >>> M-bit, or More bit, when set indicates that this is not the last DDP to be sent.
      - >>> MS-bit, or Master/Slave bit, is set in the DDP originated by the master.
  - > LSR – Link State Request packets (type 3)
  - > LSU – Link State Update packets (type 4)
  - > LSAck – Link State Acknowledgement packets (type 5)

- All LSAs sent in update packets must be individually acknowledged by one of two means:
  - > Explicit Acknowledgment – A Link State Acknowledgment packet containing the LSA header is received.
  - > Implicit Acknowledgment – An update packet that contains the same instance of the LSA is received.

- Do not confuse LSA (Link State Advertisement) with LSAck (Link State Acknowledgement).

Router Types

- All OSPF routers will be one of four router types:
  - > Internal – Routers whose interfaces all belong to the same area. These routers have a single link state database.
  - > Backbone – Routers with all interfaces attached to the backbone.
  - > ABR – Connect one or more areas to the backbone and act as a gateway for inter-area traffic.
    - Have at least one interface, each belonging to the backbone and must maintain separate link state databases for each of the connected areas.
  - > ASBR – A gateway to an external network. It injects routes into the OSPF domain that were learned (redistributed) from another external protocol/network.

LSA (Link State Advertisements)

- LSA is the OSPF data structure used to describe topology information.
- LSAs are aged as they reside in the link state database.
- MaxAge (1 hour) is the time, if reached, when LSAs are flushed from the OSPF domain.
- LSRefreshTime (every 30 min)- The router that originated the LSA floods a new copy of the LSA with an incremented sequence number and an age of zero.

- LSA Types

1- Router LSAs
> Are produced by every router for all its own connected interfaces.
> List all of a router's links, or interfaces, the state and outgoing cost of each link, and any known OSPF neighbors on the link.
> Have intra-area flooding scope.
> Describe the intra-area routes (displayed as 'O' routes in the RIB).
> Can be seen with "show ip ospf database router".

2- Network LSAs
> Are produced by the DR on every multi-access network.
> List all attached routers, including the DR itself.
> Have intra-area flooding scope.
> Identify the designated routers on a segment.
> Can be seen with "show ip ospf database network".

3- Network Summary LSAs
> Are originated by ABRs.
> Are sent into a single area to advertise destinations outside that area, but still internal to the OSPF autonomous system.
> Default routes external to the area, but internal to the OSPF autonomous system, are also advertised by LSA type 3.
> Have inter-area flooding scope.
> Describe the inter-area routes (displayed as 'O*IA' routes in the RIB).
> Can be seen with "show ip ospf database summary".

4- ASBR Summary LSAs
> Are originated by ABRs.
> Are identical to network summary LSAs, except that the destination they advertise is an ASBR, not a network.
> Have inter-area flooding scope.
> Describe which router is doing the redistribution.
> Can be seen with "show ip ospf database asbr-summary".

5- AS External LSAs
> Are originated by ASBRs.
> They advertise either a destination external to the OSPF autonomous system, or a default route external to the OSPF autonomous system.
> AS external LSAs are the only LSA type that are not associated with a particular area.
> An OSPF external route cannot use another OSPF external route as its next hop.
> Autonomous system-wide flooding scope.
> Describe what routes were redistributed (displayed as 'O*E1' or 'O*E2' routes in the RIB).
> Can be seen with "show ip ospf database external".

6- MOSPF
> Cisco routers do not support LSA Type 6 (MOSPF) and generate syslog messages if such packets are received.
> It might be necessary to configure a router to ignore these packets and thus prevent a large number of syslog messages
> Configured with "ospf ignore lsa mospf".

7- NSSA External LSAs
> Are originated by ASBRs within NSSAs (Not-So-Stubby Areas).
> Similar to an AS External LSA, except NSSA External LSAs are flooded only within the NSSA in which each was originated.

```
     > Describe redistributed routes within a NSSA area (displayed as 'O*N1' or 'O*N2' routes in the RIB).
     > Can be seen with "show ip ospf database nssa-external".

 10- Opaque LSAs
     > Used to add extensions to OSPF, such as traffic engineering parameters for MPLS networks.

- OSPF Link-State Database Overload Protection with MAX-LSA
 > Allows number of non-self-generated LSAs for a given OSPF process to be limited.
 > Used to prevent excessive LSAs generated by other routers in the OSPF domain from substantially draining the CPU and memory
     resources of a router.
 > Configured with "max-lsa".

- OSPF LSA Throttling
 > Provides a dynamic mechanism to slow down LSA updates in OSPF during times of network instability.
 > Also allows faster OSPF convergence by providing LSA rate limiting in milliseconds.
 > Configured with "timers throttle lsa all".

-----------
 COMMANDS
-----------
# sh ip ospf traffic                        - Shows OSPF traffic statistics
# sh ip ospf statistics [detail]            - Shows the SPF calculation details, including time, reason and type
                                            - Reason field descriptors:
                                            (R) A change in a router LSA (type-1) occurred
                                            (N) A change in a network LSA (type-2) occurred
                                            (SN) A change in a summary network LSA (type-3) occurred
                                            (SA) A change in a summary ASBR LSA (type-4) occurred
                                            (X) A change in an external (type-7) LSA occurred

# sh ip ospf database database-summary      - Shows the number of LSAs in a link-state database by area and by LSA type
# sh ip ospf database [router|netw|sum|asbr-sum|ext|nssa-ext]
                                            - Shows a list of the different LSAs in a link-state database

#router ospf {pid}                          >>> LSA Timers and Pacing <<<
 #timers pacing lsa-group {seconds}         - Allows more LSAs to be grouped together before being flooded (def = 240 sec)
 #timers pacing flood {milliseconds}        - Control the rate at which LSA updates occur (def = 33 ms)
 #timers throttle spf {delay} {holdtime}- Changes the delay time between receiving a topology change and SPF calculation
 #timers throttle lsa all {start} {hold} {max}  - Sets the rate-limiting intervals (in milliseconds) for LSA generation
                                            - {start-interval}: (def = 0 ms)
                                            - {hold-interval}: (def = 5000 ms)
                                            - {max-interval}: (def = 5000 ms)

 #max-lsa {max-no} [threshold-%] [warning-only] [ignore-time] [ignore-count] [reset-time]
                                            - Limits the number of non self-generated (LSAs) in the OSPF LSDB
                                            - {max number}: Number of non-self-generated LSAs
                                            - [threshold]: Percentage at which a warning message is logged (def = 75%)
                                            - [warning-only]: OSPF process never enters ignore state (def = disabled)
                                            - [ignore-time]: Time to ignore neighbors after the limit's exceeded. (def = 5min)
                                            - [ignore-count]: Number of times consecutively to enter ignore state (def = 5)
                                            - [reset-time]: Time before ignore count gets reset (def = 10 min)
 #ospf ignore lsa mospf                           - Ignores MOSPF LSA packets, stops generating syslog messages
 #neighbor {ip} database-filter all out           - Blocks the flooding of OSPF LSA packets only to a specific neighbor

#interface s0/0
 #ip ospf database-filter all out                 - Blocks the flooding of OSPF LSA packets out an interface


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================*
  Area Types
*====================*
- LSA filtering is done in two ways:
 > Area Types.
 > LSA type 3 (Network Summary) filtering see filtering below.

- When only a single area network is used, it does not have to be area 0.
- The rule is that all areas must connect to the backbone; therefore, a backbone area is needed only if there are more than one area.

> Stub Areas
      > Type 4 ASBR Summary LSAs and 5 AS External LSAs are not flooded into stub areas.
      > Still receive type 3 (Network Summary) LSAs.
      > ABRs at the edge of a stub area use type 3  LSAs to advertise a single default route (0/0) into the area
            for destinations external to the AS.
      > The ABR will advertise this default route with a cost of 1.
      > This default cost can be changed with the "area default-cost".
      > Is configured on ALL routers in the stub area with "area stub".
      > Stub area restrictions:
            >> All routers in a stub area must have identical link-state databases, and agree to be stubs.
            >> To ensure this condition, all stub routers will set a flag (the E-bit) in their hello packets to 0.
                  They will not accept hellos with E=1 (if the E-bit = Evil-bit, then Stub-Area = Holy-Area).
            >> Virtual links cannot be configured within, nor transit, a stub area.
            >> No router within a stub area can be an ASBR or perform any type of redistribution, including static and connect.

> Totally Stubby Areas
      > Use a default route to reach not only destinations external to the autonomous system but also
            all destinations outside the area.
      > The ABR of a totally stubby area will block all type 3 LSAs with the exception of a single type 3 LSA
            advertising a default route (0/0).
      > Configured with "area stub no-summary", which is necessary only at the ABR/s; the internal routers
            use the standard stub area configuration.

> NSSA (Not-So-Stubby Areas)
      > Area that allow redistribution while retaining the characteristics of a stub area to the rest of the AS.
      > Type 4 and 5 LSAs are not allowed, but redistributed AS-external routes are allowed.
      > The ASBR in an NSSA will originate type 7 LSAs to advertise these external destinations.
      > These NSSA external LSAs are flooded throughout the NSSA but are blocked at the NSSA ABR.
      > The NSSA ASBR has the option of setting or clearing the P-bit.
      > If the NSSA's ABR receives a type 7 LSA with the P-bit set to 1, the type 7 LSA translates into a type 5 LSA
            before being flooded to other areas.
      > If the P-bit is set to 0, no translation will take place and the destination in the type 7 LSA will
            not be advertised outside of the NSSA.
      > If an NSSA has multiple ABRs, the ABR with the highest router ID will do the LSA 7 to 5 conversion.
      > Configured on ALL routers in the NSSA area with "area nssa".
      > Biggest difference to a stub area is that redistribution is allowed and no default route is sent into the area.
      > With NSSA, the ABR does not automatically originate a default route.
      > To originate a default route (0/0) into a NSSA area use the command "area nssa default-originate".

> Totally NSSA
      > Is the same as an NSSA area but additionally blocks type 3 summary LSAs
      > Type 3, 4 and 5 LSAs are not allowed, but redistributed AS-external routes are allowed.
      > The ASBR in an NSSA will originate type 7 LSAs to advertise these external destinations.
      > An ABR defines an NSSA as totally stubby and originates a default as 'O*IA'.
      > Configured with "area nssa no-summary", which is only necessary  at the ABR; the internal routers
            use the standard NSSA area configuration.

- All routers in a STUB or NSSA must agree on the STUB or NSSA flag. It is the ABR(s) of the stub or NSSA area
      that determines if it is totally-stubby or totally-NSSA by adding the keyword 'no-summary' onto the "stub/nssa" command.
- The ABR generates the type 4 LSA. If an area is configured as a stub area, the ABR filters the type 5 LSAs(generated by the ASBR)
      and does not generate a type 4 LSA. So, technically, an OSPF stub configuration explicitly filters type 5 LSAs, but it
      implicitly filters type 4 LSAs since there is no need for the ABR to generate type 4 LSAs.

- When an ABR is also an ASBR and is connected to a NSSA, the default behavior is to advertise the
      redistributed routes into the NSSA.
 > This redistribution can be turned off by adding the 'no-redistribution' keyword to the "area nssa" command.

- Suppress OSPF Forwarding Address in Translated Type-5 LSAs
 > This is used when an NSSA ABR translates type 7 LSAs to type 5 LSAs. 0.0.0.0 must be used as the forwarding address instead
      of the address specified in the type 7 LSA.
 > Routers which are configured not to advertise forwarding addresses into the backbone will directly forward traffic to the
      translating NSSA ASBRs.
```

```
-----------
 COMMANDS
-----------
 #router ospf {pid}
  #area 1 default-cost {cost}                    - Changes the cost of the default route advertised by the ABR. (default = 1)
                                                 - Copy descretly owned by Kane  Bagwell
  #area 1 stub                                   - Configures attached area 1 to be a stub area
                                                 - It is configured on all area routers
                                                 - Shows the default route in the routing table as 'O*IA 0.0.0.0/0'

  #area 2 stub no-summary                        - Configures attached area 2 to be a totally stubby area
                                                 - It is only configured on ABRs
                                                 - Shows the default route in the routing table as 'O*IA 0.0.0.0/0'

  #area 3 nssa                                   - Configures attached area 3 to be not so stubby area
                                                 - It is configured on all area routers
                                                 - NO default route is automatically generated
```

```
#area 4 nssa default-information-originate    - Configures attached area 4 to be a nssa
                                              - It is only configured on ABRs to generate the default
                                              - Shows the default route in the routing table as 'O*N2 0.0.0.0/0'

#area 5 nssa no-summary                       - Configures attached area 5 to totally-nssa
                                              - It is only configured on ABRs
                                              - Shows the default route in the routing table as 'O*IA 0.0.0.0/0'

#area 6 nssa no-redistribution no-summary     - Configures attached area 6 to totally-nssa with default redistribution disabled
                                              - Shows the type 3 default route in the routing table as 'O*IA 0.0.0.0/0'

#area 7 nssa no-redistribution default-information-originate
                                              - Configures a nssa, allowing type 3, blocking type 4, 5 and 7
                                              - Shows the type 7 default route in the routing table as 'O*N2 0.0.0.0/0'

#area 8 nssa translate type7 suppress-fa      - Suppresses the inclusion of a forwarding address when translated into type 5 LSAs

##area type options explained###              >> [stub] blocks type 4 and type 5 LSAs
                                              >> [no-summary] blocks type 3 LSAs except the default route type 3 LSA
                                              >> [nssa] blocks type 4 and type 5 LSAs, but allows type 7 redistribution
                                              >> [no-redistribution] blocks type 7 LSAs
```

-=-

======================

Filtering

======================

- Filtering may only occur between areas by RFC standard: 'All routers within an area must have the same link-state database'.
- Different ways to filter traffic:
 > With the "filter-list" command.
 > With the "distribute-list" command referencing an ACL/prefix-list/route-map.
 > With the "distance" command.
 > With the "area range" command (refer to the next section below).
 > With the "summary-address" command for external prefix filtering (refer to the next section below).

- The ABRs can filter network addresses being advertised by type 3 LSAs either into or out of an area.
```
 > In-lists        - Filter LSAs before they are sent into an area.
 > Out-lists       - Filter LSAs leaving an area to prevent those LSAs from entering any other areas attached to that router.
```

- Distribute-List
 > Note that distribute-lists ONLY block routes from entering the LOCAL RIB, they do not stop LSA propagation.
 > Using a distribute-list out has NO effect within an OSPF area since all routers in a area must have the same database.
 > Using a route-map the following 'match route-type' criteria can used with OSPF:
```
      >>  External        OSPF external route types E1 and E2.
      >>  Internal        OSPF internal routes (includes OSPF intra/inter-area)
      >>  Local           OSPF routes locally generated on the router.
      >>  NSSA-external   OSPF NSSA-external route type N1 and N2.
```

```
-----------
 COMMANDS
-----------
#ip prefix LIST1 seq 10 deny 10.5.1.0/24        - Matches 10.5.1.0/24 to be denied
#ip prefix LIST1 seq 20 permit 0.0.0.0/0 le 32  - Permits everything else

#router ospf (pid)
 #area 0 filter-list prefix LIST1 out           - Filters traffic leaving out of (from) area 0, matching the prefix-list
                                                - This will apply to all areas that the local router is connected to
 #area 25 filter-list prefix LIST1 in           - Filters traffic sent into area 25, i.e. don't send 10.5.1.0
                                                - Does the same as above, but only for area 25

 #distribute-list {acl|prefix|route-map} in     - This filter applies ONLY to routes entered into the local RIB
 #distribute-list prefix LIST1 in               - This stops 10.5.1.0 from entering the RIB, but it's still in LSA-DB
 #distance 255 10.5.1.5 0.0.0.0 99              - Assigns admin distance 255 for routes matching ACL-99 from source 10.5.1.5
 #distance ospf {ext|inter-area|intra-area}     - Changes the distance of OSPF route types


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Summarization
*=====================*
- Best practice dictates that a non-backbone area's addresses should be summarized into the backbone by the area's own ABR.

- Two types of address summarization are supported by OSPF.
 > Inter-Area Summarization
      >> Used for summarization of internal OSPF area routes at ABRs.
      >> A route to Null0 will be entered into the routing table automatically, but can be disabled with "no discard-route".
      >> The "area range" command specifies the area to which the summary address belongs.
      >> The default behavior of the "area range" command is to advertise more specifics along with the specified summary
            but this can be suppressed with the 'no-advertise' keyword.
      >> Summarizes type 3 LSAs.

 > External Route Rummarization
      >> Allows a set of external addresses to be redistributed into an OSPF domain as a summary address at the ASBRs.
      >> Configured with "summary-address" command on the ASBRs.
      >> Specifics of the specified summary address will be suppressed.
      >> Summarizes type 5 and 7 LSAs.


-----------
 COMMANDS
-----------
 #no discard-route                              - Disables creation of the Null route when using the area range command
 #area 15 range 10.0.0.0 255.0.0.0 [advertise] [not-advertise] [cost]
                                                - Specifies the area to which the summary address belongs
                                                - [advertise] Advertise more specifics (default)
                                                - [not-advertise] Does NOT advertise more specifics
                                                - [cost] User specified metric for this range
 #summary-address 150.1.60.0 255.255.255.0 [not-advertise]
                                                    - Summarizes type 5 and type 7 LSAs
                                                    - More-specifics which are within the range will be suppressed
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Stub Router Advertisement
*==========================*
```

- Do not confuse this with STUB AREAS.
- A stub router advertisement is an update sent with a maximum metric set.
- Two main benefits of OSPF stub router advertisement
  > Allow a new router to be brought into the OSPF domain without immediately routing traffic through it.
  > Allow a router to be reloaded gracefully by having the rest of the OSPF domain route around the router that is being reloaded.

- Advertises a maximum metric for all routes that the particular router does not originate.
- Also the feature can be used to allow the router to advertise a maximum metric until the BGP routing table converges.
- A typical scenario for the use of stub router advertisement is when there are multiple links between two areas, and one link should only be used as a last resort.

- Three different configuration sets:
 1- To configure an OSPF router to advertise a maximum metric during startup.
 2- To configure an OSPF router to advertise a maximum metric until BGP routing tables converge.
 3- To configure an OSPF router to advertise a maximum metric for a graceful shutdown or removal from the network.

```
CONFIG-SET 1: Configuring Max-Metric advertisements on startup
+-----------------------------------------------------
|     router ospf 1
|      max-metric router-lsa on-startup {sec}   - Advertises a maximum metric during startup for announce-time = sec
|                                               - There is no-default, (value = 5-86400 sec)
|

CONFIG-SET 2: Configuring Max-Metric advertisements until routing tables converge
+-----------------------------------------------------
|     router ospf 1
|      max-metric router-lsa on-startup {sec} wait-for-bgp
|                                               - {sec} Time that router-LSAs are originated with max-metrics
|                                               - [bgp] Lets BGP decide when to originate router-LSA with normal metric
|                                               - (def = 600 sec)
|

CONFIG-SET 3: Configuring Max-Metric advertisements for a graceful shutdown
+-----------------------------------------------------
|     router ospf 1
|            max-metric router-lsa              - Configures OSPF to advertise a max-metric. This causes neighbors to
|                                                   select an alternate path before the router is shutdown
|
```

COMMANDS

```
#max-metric router-lsa [summary-lsa | include-stub | external-lsa | on-startup]
                                                  - Sets a maximum metric for self-originated router-LSAs
                                                  - [summary-lsa] Overrides summary-lsa metric with max-metric value
                                                  - [include-stub] Sets maximum metric for stub links in router-LSAs
                                                  - [external-lsa] Overrides external-lsa metric with max-metric value
                                                  - [on-startup] Sets maximum metric temporarily after reboot
```

Passive-Interface

- The passive-interface with OSPF will prevent hello packets from exiting an interface and prevent the device from forming any adjacencies out of the specified interface.
- This works differently to distance vector protocols like RIP, where routes will still be received, but not sent.
- To get the same 'passive-interface' effect as distance vector protocols in OSPF,(i.e. receive routes but don't send routes) use: "ip ospf database-filter all out" under the interface.

COMMANDS

```
# sh ip ospf interface                            - Passive-interfaces indicated by 'No Hellos' in output

#router ospf {pid}
 #passive-interface {int}                         - Prevents hello sent out an interface
                                                  - Prevents forming of adjacencies out that interface
#interface s0/0
 #ip ospf database-filter all out                 - Block the flooding of OSPF LSA packets out the interface
                                                  - Filtering the outbound updates breaks RFC standards
```

Originating a Default Route

- A default route is announced as an IP prefix 0.0.0.0/0 in OSPF.
- Contrary to many other routing protocols, the default route cannot be redistributed into OSPF. It must be configured manually.
- A default route can be inserted into OSPF only as an external or inter-area (summary) route, not as an intra-area route.

- There are different methods to originate a default route within OSPF:
 > Unconditional default route.
 > Conditional default route.
 > Conditional default route with a route-map.
 > OSPF stub areas.
 > OSPF NSSA default routes.

- Unconditional OSPF Default Route
 > This advertises a default route into the OSPF domain, regardless of whether the local router can reach areas outside the OSPF domains, or not.
 > With no additional configuration options, the default route is advertised as an External Type 2 (E2) route with metric 1.
 > Configured with "default-information originate always" under the OSPF process.

- Conditional OSPF Default Route
 > Configured with "default-information originate" but without the 'always' keyword.
 > This advertises a default route into the OSPF domain, but only if the advertising router has a non-OSPF default route in its routing table.
 > The non-OSPF default route could be any of the following:
   >> A static default route with the next-hop pointing outside the OSPF domain.
   >> A static default route based on IP SLA measurements (example: http://tinyurl.com/ospf-ipsla).
   >> Or a BGP advertised default route.
 > The "default-information originate" command without the always option is functionally equivalent to redistributing a default route into OSPF.
 > With no additional configuration options, the default route is advertised as an E2 route with a metric of 1.
 > The 'metric' keyword could be used to change the default route's default metric of 1.

CONFIG-SET: Conditional OSPF Default Route with a Non-Default Cost

```
+------------------------------------------------------------------
|     ip route 0.0.0.0 0.0.0.0 serial0.1           - Route via the interface connecting to a upstream ISP
|     !
|     router ospf 1
|      default-information originate metric 10     - Originates a default route provided the above static is in
|                                                        the routing-table. If interface serial0.1 goes down,
|                                                        OSPF will stop advertising the default route
|                                                  - Default route is also advertised with a cost of 10
```

- Conditional OSPF Default Route with a Route-Map
 > The advertisement of a default route into OSPF could also be subject to specific routing information in the routing table.
 > For this function a route-map is used. Thus whenever a route in the IP routing table matches the conditions specified in the route-map, the default route will be advertised into OSPF.
 > Because the specified route-map checks the entries in the IP routing table, it may ONLY match on IP prefixes, next-hops and metrics. BGP attributes may not be used.
 > Configured with "default-information originate route-map {name}".

CONFIG-SET: Conditional OSPF Default Route with a Route-Map

```
+--------------------------------------------------------------
|     ip prefix-list UPLINKS permit 10.5.1.0/24    - Matches uplink 1
|     ip prefix-list UPLINKS permit 10.5.2.0/24    - Matches uplink 2
|     !
|     route-map CON
|      match ip address prefix UPLINKS             - References either uplink to be present in the routing table
|     !
|     router ospf 1
|      default-information originate route-map CON - Advertises a default route provided one of the two uplinks is
|                                                        present in the routing table
```

```
- OSPF Stub Area's Default Route
 > Refer to area section for more detail.
 > To ensure end-to-end connectivity between routes in stub areas and external destinations, the area ABRs by default
      originates a default route into stub areas. These routes are advertised as inter-area (summary) routes.
 > Unless configures otherwise, the default routes are advertised into the stub area with OSPF metric of 1.
 > When multiple exit points out of a stub area exist, the routers will select the nearest ABR to reach external destinations.
 > If desired, the inter-area default route metric can be changed with the area default-cost router configuration command.

CONFIG-SET: OSPF Stub Area's Default Route using a Non-Default Cost
+--------------------------------------------------------------------
|     router ospf 1
|      area 1 stub                               - Configures area 1 as a stub, and advertises a default route
|      area 1 default-cost 300                   - Changes the cost of the stub default route to 300



- OSPF NSSA's Default Routing
 > Cisco routers don't advertise external default routes into an NSSA area, even when configured
      with "default-information originate always".
 > The ABRs can be configured to advertise the OSPF default route using one of the following options:
      >> Manually advertise a type-7 (NSSA external) default route into the NSSA area.
            >>> Configured with "area nssa default-information-originate".
      >> Configure an NSSA as a totally NSSA and generate an inter-area (type 3) external route.
            >>> Configured with "area nssa no-summary".



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   Path Selection
*======================*
- Each OSPF route entry is classified according to a destination type. The destination type will be either network or router.
 > Network entries are the addresses of networks to which packets can be routed. These are destinations which are candidates for
   insertion into the routing table.
      >> Seen with "sh ip route ospf".
 > Router entries are routes to ABRs and ASBRs. This information is kept in a separate internal OSPF router table.
      >> Seen with "sh ip ospf border-routers".

- OSPF Route Table Lookups:
 1- Longest match
 2- Most to least preferred path type:
      >> O        - Intra-area paths are to destinations within one of the router's attached areas.
      >> O IA           - Inter-area paths are to destinations in another area but within the OSPF AS.
      >> E1 (N1)  - Paths to external destinations. External cost + cost to ASBR.
      >> E2 (N2)  - Paths to external destinations. Only external cost to destination is used. !!DEFAULT!!
 3- Use lowest cost metric unless equal-cost paths exist.

- OSPF external routes are classified as E2 routes by default.
- E1 and N1 are cumulative metrics, combining the ASBR advertised cost and internal OSPF cost to the ASBR.
      E1 and N1 are often referred to as 'metrics that increase hop-by-hop'.
```

```
- E2 and N2 are static metrics as advertised by ASBR.
- Use E1 metrics when packets should exit the network at the closest exit point.
- Use E2 metrics when packets should exit the network at the closest point to the external destination.

- Cost is the OSPF metric expressed as a 16-bit integer in the range of 1 to 65535.
- Cisco uses a default cost of 10^8/BW expressed in whole numbers, i.e. 100MB interface gets a metric of 1, 10Mb = 10, etc.
      BW is the configured bandwidth of the interfaces and 10^8 is the reference bandwidth.
- The reference bandwidth of 10^8 (100MB) creates a problem for some modern media with bandwidths higher than 100M.
- To remedy this, the command "auto-cost reference-bandwidth" may be used to change the default reference.
- Default Cost = Reference-BW/Interface-BW.

- The OSPF cost can be modified with:
 > Interface "bandwidth".
 > Interface "ip ospf cost".
 > Process "auto-cost reference-bandwidth".
 > Process "neighbor x.x.x.x cost" on point-to-multipoint non-broadcast areas.

- iSPF (OSPF Incremental SPF)
 > Incremental SPF is more efficient than the full SPF algorithm allowing slightly faster convergence.
 > Incremental SPF allows the system to recompute only the affected part of the SPF tree.
```

COMMANDS

```
# sh ip ospf border-routers                      - Shows the internal OSPF table. Entries are routes to ABRs and ASBRs

#interface s0/1
 #bandwidth {kbps}                               - Changes the default interface bandwidth numerical integer
 #ip ospf cost {value}                           - Changes the outgoing interface cost for packets

#router ospf {pid}
 #auto-cost reference-bandwidth     {mpbs}       - Remedies the Cisco default reference bandwidth problem (def = 100 mbps)
 #neighbor {ip} [priority {pri}] [cost {cost}]   - Changes the cost for routes received from the specified neighbor
 #ispf                                           - Enables iSPF
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
```

Authentication

```
*======================*
- If area authentication is configured, it must be configured for the entire area.
- Don't forget about virtual-links. Virtual-link always have one interface in Area 0.
- Interface passwords do not have to match throughout the area, but must be the same between neighbors.
- By default OSPF uses NULL authentication.
- OSPF supports the following authentication types:
 > (type 0) Null authentication
 > (type 1) Clear-text passwords
 > (type 2) MD5 cryptographic checksums
```

- Authentication keys are locally significant to an interface and therefore may differ on a per interface basis.
- When doing changes to the keychain, it is recommended to first remove the config of the interface.

- Virtual-Link Authentication can be enabled in the following two ways:

```
      #area {id} authentication [message-digest]
      #area {id} virtual-link router-id authentication [message-digest | null]
```

```
-----------
 COMMANDS
-----------
# sh ip ospf interface {int}                   - Shows if message-digest is configured
# sh ip ospf | i Area                          - To see if authentication is enabled for the area (with capital 'A')

#router ospf {pid}
 #area 10 authentication                       - Enables type 1 authentication for area 10
 #area 20 authentication {message-digest}      - Enables type 2 MD5 authentication for area 20
 #area 30 virtual-link 1.1.1.1 auth {key}      - Enables type 1 authentication for the virtual-link
 #area 40 virtual-link 2.2.2.2 message-digest-key {key-id} md5 {key}
                                               - Enables type 2 MD5 authentication on a virtual-link
#interface serial1
 #ip ospf authentication null                  - Enables type 0 authentication. Thus no authentication needed on the interface

#interface serial0
 #ip ospf authentication                       - Enables type 1 interface authentication
 #ip ospf authentication-key {key}             - Specifies the type 1 key string

#interface serial2
 #ip ospf authentication message-digest        - Enables type 2 MD5 interface authentication
 #ip ospf message-digest-key {id} md5 {key}    - Specifies the type 2 MD5 key string
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
   OSPF Demand Circuit
*========================*
```

- Demand circuit suppresses the hello and LSA refresh functions. Historically used dial-up links do not have to stay UP.
- OSPF brings up a demand circuit to perform the initial database synchronization and subsequently to only flood the LSAs in which certain changes have occurred.
- These LSA changes are:
 > A change in the LSA options field.
 > A new instance of an existing LSA is received in which the age is MaxAge.
 > A change in the length field of the LSA header.
 > A change in the contents of the LSA, excluding the 20-octet header, the checksum, or the sequence number.

- Because no periodic hellos are exchanged, OSPF makes a presumption about reachability.
- OSPF demand circuit sets the "do not age" flag on all learned LSAs and will only send updates when there is a change in the OSPF topology.
- The command should be configured on point-to-point links and is required only on one side.
- If the router is part of a point-to-multipoint topology, only the multipoint end must be configured with this command.

```
- Configured with "ip ospf demand-circuit".
- Changes to the interface and neighbor state machines and to the flooding procedure:
 > MaxAge = DoNotAge
 > A flag known as the DC-bit (Demand Circuit Bit) is added to all LSAs the demand circuit originates.
 > The DoNotAge bit is set on LSAs advertised out, therefor the interface and the LSAs are not refreshed unless they change.

-----------
 COMMANDS
-----------
#ip ospf demand-circuit                          - Configures the connected interface to the demand-circuit
#ip ospf flood-reduction                         - The DoNotAge bit is set on LSAs advertised out of the interface



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   Troubleshooting OSPF
*=========================*
- When examining an individual router's configuration, consider the following:
 > Are the necessary interfaces in a UP,UP state (not admin shut)?                         # sh ip int brief
 > Do all the interfaces have the correct addresses and masks?                             # sh int | i Inter|line
 > Do the network area statements and interface IP addresses correlate?                    # sh int | i Inter|line|network
 > Do the network area statements have the correct inverse masks?                          # sh ip ospf int brief
 > Are the network area statements putting the interfaces into the correct areas?          # sh ip ospf int brief
 > Are any interfaces wrongly in passive mode, due to "passive-interface default"          # sh ip ospf int | i line|Hello
 > Does each router have the correct router ID? Any duplicates on the network?             # sh ip ospf | i ID
 > If address summarization is configured, is it applied to the correct areas?             # sh run | i area range|summary-add

- When examining an area-wide problem, consider the following by looking at the design:
 > Is the backbone (Area 0) one contiguous domain?                                         - Not segregated?
 > Are all areas connected to Area 0?                                                      - Directly or in-directly
 > Are all routers in an area configured as the same area type?                            - Normal, Stub, or NSSA
 > Are all ABRs configured correctly with the correct role?                                - Totally-stub, or totally-NSSA
    Remember with multiple NSSA ABRs, only the router with the highest RID does the conversion!
 > Is there a virtual link that transits or is configured within a stub area?              - If so, configure GRE tunnel instead
 > Is a default summary LSA present allowing exit out of an area for unknown subnets/ASs? - A NSSA needs a manual default route
 > Does an external LSA exist to leave the OSPF domain?                                    # sh ip ospf data external
 > Is the forwarding address known as an internal OSPF route? It must be.                  # sh ip route {fa-ip}
 > Is the forwarding address reachable?                                                    # ping {fa-ip}

- When examining adjacencies (or the lack thereof), consider the following:
 > It could be helpful to log the neighbor adjacency changes.                              #ospf log-adjacency-changes
 > Is there layer2 connectivity and layer3 reachability?                                   # ping {neighbor-ip}
 > Are hellos being sent from both neighbors and received by both?                         # debug ip ospf hello
    >> If not check the network statements and interfaces addresses.                       # sh ip ospf int brief
    >> Any interfaces wrongly configured as "passive-interface"?                           # sh run | i passive
 > If different network types, are they compatible?                                        # sh ip ospf int | i line|Type
 > Are the hello/dead timers the same between neighbors?                                   # sh ip ospf int | i line|Dead
 > Are the optional capability value the same between neighbors?                           # sh ip ospf neighbor detail | i Option
 > Are the interfaces configured on the same subnet (this excludes point-to-point links)? # sh ip ospf int brief
 > Is a router attempting to form an adjacency with another's secondary address?          # sh run | i netw|area
 > Are any access-lists blocking OSPF protocol 89?                                        # sh ip interface | i line|list
 > If the neighbor is a switch, are the MTU values are the same?                          # debug ip ospf adj
 > If suspecting that the adjacencies are unstable or as a last resort, use.              # debug ip ospf adj

- When using frame-relay, consider the following:
 > Do multipoint NBMA interfaces have static layer3-to-layer2 mappings?                   # sh run | i frame.*map
 > Is frame-relay broadcast replication enabled where necessary?                          # sh run | i frame.*broadcast
 > In a hub-spoke scenario are any of the spokes blackhole DRs (spoke = 0 priority)?      # sh ip ospf interface {int} | i ID

- When troubleshooting authentication issues, consider the following:
 > Are all routers within an area configured to use authentication?                       # sh ip ospf | i Area
 > Is the authentication type the same between neighbor interfaces?                       # sh ip ospf int {int} | i auth|line
 > With clear-text auth, are the passwords the same between neighbor interfaces?          # sh run | i auth.*key
 > With MD5 authentication, are the digest-keys the same between neighbor interfaces?     # sh run | i digest-key
 > Do all the virtual links also have authentication configured?                          # sh run | i virtual-link
 > If Area 0 has authentication configured, then virtual links require authentication too.
 > To see the cause of authentication failures, use:                                      # debug ip ospf adj

- Link-state database problems. (All databases must be the same for each area)
 > Is the local router generating the expected LSAs?                                      # sh ip ospf database self-originate
 > Is the local router receiving the expected LSAs from a neighbor?                       # sh ip ospf database adv-router {ip}
 > Are any filters configured to deny LSAs being sent into an area?                       # sh run | i filter-list
 > Are any distribute lists configured to deny entry in the local RIB?                    # sh run | i distribute-list
 > Is summarization the cause of LSAs not being seen?                                     # sh run | i area range|summary-add
 > Do all the routers in an area have the same number of LSAs?                            # sh ip ospf database database-summary
      >> If not, are any interfaces filtering LSAs from being sent out?                   # sh run | i database-filter
 > Do the checksums for every LSA in the databases match between routers?                 # sh ip ospf database
 > Do any LSAs have a higher than others sequence number (look at Seq#)?                  # sh ip ospf database
      >> This could point to an unstable link, caused by frequent LSA advertising.        # sh int {int} | i error|drops
      >> Multiple LSAs with high sequence numbers could indicate a neighbor issue.        # sh ip ospf neighbor detail | i Neighbor
 > Have there been many SPF calculations? What triggered these events?                    # sh ip ospf statistics
 > Have you checked the memory-and CPU-utilization on the routers?                        # sh process cpu history
                                                                                          # sh process memory sorted
- When doing redistribution, consider the following:
 > Is the 'subnets' keyword used in the statement?                                        # sh run | i redistribute.*subnets
 > For BGP redistribution,
      >> If needed, was the 'external' keyword specified?                                 - Not done by default
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
      OUTPUT-101
*======================*
-----------------------------------------------------
      # sh ip ospf interface serial1.738
-----------------------------------------------------
Serial1.738 is up, line protocol is up
   Internet Address 10.5.21.21/30, Area 7
   Process ID 1, Router ID 10.5.30.70, Network Type POINT_TO_POINT, Cost: 781
   Transmit Delay is 1 sec, State POINT_TO_POINT,
   Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:07
   Neighbor Count is 1, Adjacent neighbor count is 1
      Adjacent with neighbor 10.5.30.77
   Message digest authentication enabled
      Youngest key id is 10

-----------------------------------------------------
      # sh ip ospf interface ethernet1
-----------------------------------------------------
Ethernet1 is up, line protocol is up
   Internet Address 10.5.32.4/24, Area 78
   Process ID 1, Router ID 10.5.30.70, Network Type BROADCAST, Cost: 10
   Transmit Delay is 1 sec, State DROTHER, Priority 1
   Designated Router (ID) 10.5.30.254, int address 10.5.32.2
   Backup Designated router (ID) 10.5.30.80, int address 10.5.32.1
   Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:01
   Neighbor Count is 5, Adjacent neighbor count is 2
      Adjacent with neighbor 10.5.30.80 (Backup Designated Router)
      Adjacent with neighbor 10.5.30.254 (Designated Router)
   Message digest authentication enabled
      Youngest key id is 10

- IP Address/Mask       - OSPF packets originated from this interface will have this source address.
- Area ID               - OSPF packets originated from this interface will have this Area ID.
- Process ID            - This Cisco-specific feature. Cisco routers are capable of running multiple OSPF processes and use the
                              Process ID to distinguish them.
- Network Type          - The type of network to which the interface is connected- broadcast, point-to-point, NBMA,
                              point-to-multipoint, or virtual link.
- Cost                  - The outgoing cost for packets transmitted from this interface.
- InfTransDelay         - The seconds by which LSAs exiting the interface will have their ages incremented (default = 1 sec)
- State                 - The functional state of the interface.
- Router Priority       - Priority is only displayed on multi-access links.
- Wait Timer            - The time to wait for a DR and BDR to be advertised in a neighbor's hello packet before beginning a
                              DR and BDR selection.
- Retransmit Interval   - The period, in seconds, the router will wait between retransmissions of OSPF packets that have
                              not been acknowledged.
```

```
-----------------------------------------------------
      # sh ip ospf database
-----------------------------------------------------
OSPF router with ID (10.5.30.50) (Process ID 1)

       Router Link States                                     - Type 1 LSAa
Link ID            ADV Router      Age   Seq#         Checksum   Link count
10.5.30.10      10.5.30.10   1010  0x80001416  0xA818     3
10.5.30.20      10.5.30.20   677   0x800013C9  0xDE18     3

      Net Link States                                         - Type 2 LSAa
Link ID            ADV Router      Age   Seq#         Checksum
10.5.17.18      10.5.30.20    677   0x800001AD  0x849A
10.5.17.34      10.5.30.60    695   0x800003E2  0x4619

      Summary Net Link States                                 - Type 3 LSAs
Link ID            ADV Router      Age   Seq#         Checksum
10.5.121.0       10.5.30.40    1231  0x80000D88  0x73BF
10.5.121.0       10.5.30.50    34    0x800003F4  0xF90D

      Summary ASB Link States                                 - Type 4 LSAs
Link ID            ADV Router      Age   Seq#         Checksum
10.5.30.12      10.5.30.40   1240  0x80000006  0x6980
10.5.30.12      10.5.30.50   42    0x80000008  0xC423

      AS External Link States                                 - Type 5 LSAs
Link ID            ADV Router      Age   Seq#         Checksum   Tag
10.83.10.0         10.5.30.60   459   0x80000D49  0x9C0B     0
10.22.85.0         10.5.30.80   1056  0x800001F7  0x6B4B     65502


- The router link states       - LSAs are generated by each router within an area for all its connected interfaces.
- The net link states          - LSAs are only created by the DR.
- The summary net link states  - Shows networks not local to the area.
- Summary ASB link states      - Shows which router is doing the redistribution.
- AS external link states      - Shows the redistributed routes.
```

```
--------------------------------------------------------
      # sh ip ospf database router 10.5.30.10
--------------------------------------------------------
      OSPF Router with ID (10.5.30.50) (Process ID 1)
                    Router Link States (Area 0)

Routing Bit Set on this LSA
LS age: 680
Options: (No TOS-capability)
LS Type: Router Links
```

```
Link State ID: 10.5.30.10
Advertising Router: 10.5.30.10
LS Seq Number: 80001428
Checksum: 0x842A
Length: 60
Area Border Router
 Number of Links: 3

  Link connected to: another router (point-to-point)
   (Link ID) Neighboring router ID: 10.5.30.80
   (Link Data) router int address: 10.5.17.9
    Number of TOS metrics: 0
     TOS 0 Metrics: 64

  Link connected to: a Stub Network
   (Link ID) Network/subnet number: 10.5.17.8
   (Link Data) Network Mask: 255.255.255.248
    Number of TOS metrics: 0
     TOS 0 Metrics: 64

  Link connected to: a Transit Network
   (Link ID) Designated router address: 10.5.17.18
   (Link Data) router int address: 10.5.17.17
    Number of TOS metrics: 0
     TOS 0 Metrics: 10
```

```
- 'Routing Bit Set on this LSA'
 > Is not a part of the LSA itself.
 > It is an internal maintenance bit used by IOS indicating that the route to the destination advertised by this LSA is valid.
 > It means that the route to this destination is a candidate for the routing table.
```

```
---------------------------------------------------------
      # sh ip ospf database database-summary
---------------------------------------------------------
            OSPF router with ID (10.5.30.50) (Process ID 1)
Area ID        router  Network  Sum-Net  Sum-ASBR  Subtotal  Delete  Maxage
0              8       4        185      27        224       0       0
4              7       0        216      26        249       0       0
5              7       0        107      13        127       0       0
56             2       1        236      26        265       0       0
AS External                                        580       0       0
Total          24      5        744      92        1445
```

```
-----------------------------------------------------------
      # sh ip ospf database external 10.5.60.0
-----------------------------------------------------------
            OSPF router with ID (10.5.5.5) (Process ID 1)
                Type-5 AS External Link States
  Routing Bit Set on this LSA
  LS age: 1672
  Options: (No TOS-capability, DC)
  LS Type: AS External Link
  Link State ID: 10.5.60.0 (External Network Number )
  Advertising Router: 10.5.6.6                                          - The router that advertised this LSA
  LS Seq Number: 80000002
  Checksum: 0x24EF
  Length: 36
  Network Mask: /24
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 0


-----------------------------------------------------
      # debug ip ospf adj                                               - Shows the OSPF neighbor states
-----------------------------------------------------
OSPF adjacency events debugging is on
OSPF: Rcv DBD from 10.5.30.70 on Ethernet0 seq 0x20E0 opt 0x2 flag 0x7 len 32 state INIT
OSPF: 2 Way Communication to 10.5.30.70 on Ethernet0,  state 2WAY
OSPF: Neighbor change Event on int Ethernet0
OSPF: DR/BDR election on Ethernet0
OSPF: Elect BDR 10.5.30.70
OSPF: Elect DR 10.5.30.175 DR: 10.5.30.175 (Id) BDR: 10.5.30.70 (Id)
OSPF: Send DBD to 10.5.30.70 on Ethernet0 seq 0xB17 opt 0x2 flag 0x7 len 32
OSPF: First DBD and we are not SLAVE
OSPF: Rcv DBD from 10.5.30.70 on Ethernet0 seq 0xB17 opt 0x2 flag 0x2 len 92 state EXSTART
OSPF: NBR Negotiation Done. We are the MASTER
OSPF: Send DBD to 10.5.30.70 on Ethernet0 seq 0xB18 opt 0x2 flag 0x3 len 72
OSPF: Database request to 10.5.30.70
OSPF: Rcv DBD from 10.5.30.70 on Ethernet0 seq 0xB18 opt 0x2 flag 0x0 len 32 state EXCHANGE
OSPF: Send DBD to 10.5.30.70 on Ethernet0 seq 0xB19 opt 0x2 flag 0x1 len 32
OSPF: Rcv DBD from 10.5.30.70 on Ethernet0 seq 0xB19 opt 0x2 flag 0x0 len 32 state EXCHANGE
OSPF: Exchange Done with 10.5.30.70 on Ethernet0
OSPF: Synchronized with 10.5.30.70 on Ethernet0, state FULL


---snip---
OSPF: Nbr 10.5.11.6 has larger interface MTU                            - Indicates two interfaces have different MTU sizes
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY

```
 ______    ______    ______
|  ___ \   / _____)  |  ___ \
| |___) ) | | ____   | |___) )
|  ___ (  | | (_  |  | _____/
| |___) ) | |____) |  | |
|______/   \_____/   |_|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-*
|       INDEX           |
*-=-=-=-=-=-=-=-=-=-=-=-*
```

- The BGP Process
- Establishing Peerings
  + Update Source
  + BGP States
  + BGP Open Message
- Authentication
- eBGP Sessions
  + eBGP Multihop
  + BGP Backdoor
  + Distance
  + Maximum-Paths
- Next-Hop Processing
  + Next-Hop Self
  + Route-Map
- iBGP Sessions
  + Route Reflection
  + Confederation
- iBGP Synchronization
  + Redistribution into IGP
  + BGP over GRE
- BGP Attributes
  + Bestpath Selection Process
  + Cisco Weight Attribute
  + Local Preference Attribute
  + AS-Path Attribute
  + MED (Multi Exit Discriminator) Attribute
  + Community Attribute
    o Standard Communities
    o Extended Communities
    o BGP Community Formats
    o Well-Known Standard Communities
    o Additive Keyword
    o Community-Lists
    o Cost Community
    o BGP Dmzlink Bandwidth
- Originating Prefixes
  + Network Statement
  + Redistribution
  + Default Information Originate
  + Neighbor Default Originate

```
      + Aggregation
            o Summary-Only
            o Suppress-Map
            o Unsuppress-Map
- Filtering
      + As-Path Filter
      + Distribute-Lists
      + Prefix Filter
- Regular Expressions
- Conditional Advertisement
- Conditional Route Injection
- Clearing BGP Sessions
- ORF (Outbound Route Filtering)
- BGP Network Migration
      + Local AS
      + Remove Private AS
- BGP Route-Maps
- BGP Route-Dampening
- Peer Groups
- Peering Templates
- Fast External Fallover
- Fast Peering Session Deactivation
- Support for Next-Hop Address Tracking
- Maximum-Prefix
- BGP Policy Accounting
- Troubleshooting BGP
      + BGP Errors
      + BGP Session Start-up Problems
      + Route Selection Issues
- OUTPUT-101
      + Clearing a TCP session
      + "sh ip bgp" Explained
      + "sh ip bgp summary" Explained


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  The BGP Process
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
         > Cisco BGP Overview

- Defined by RFC 1771.
- BGP is a path vector protocol that uses TCP port 179 for reliable transport.
- BGP has no periodic updates, it uses triggered updates (when available):
 > Every 5 sec for internal peers.
 > Every 30 sec for external peers.
```

```
- Periodic keepalives are used to verify TCP connectivity:
 > By default every 60 sec.
- Holdtime interval is the time after which a notification is sent if no keepalives have been received (default = 180 sec).
- Only the holdtime is sent in updates. Two peers will agree on the lowest holdtime value between them, and then calculate the
      keepalive value based on this holdtime value.

-----------
 COMMANDS
-----------
#router bgp {asn}                                 - Enables/enters BGP config-mode (ASN = autonomous system number)
 #bgp router-id {ip}                              - Configures the RID for BGP Process

 #bgp scan-time {scanner-interval}                - Changes the default value of the BGP scanner process (max/default = 60 sec)
                                                  - The BGP scanner walks the BGP table and confirms the reachability of next-hops
                                                  - The BGP scanner process is also responsible for conditional advertisement
                                                         checks and performing route dampening

 #timers bgp {keepalive} {holdtime}               - Changes the default BGP timer values (60sec, 180sec)
                                                  - Only the holdtime value is communicated in the BGP open message
                                                  - Smallest configured holdtime value between BGP peers are used by both peers and
                                                        used to determine the keepalive

 #neighbor {ip|peer-group} advertisement-interval {sec}
                                                  - Changes the default time interval of sending BGP routing updates
                                                  - Lowered value can improve convergence but can consume considerable resources
                                                         in a jittery network if value is too low. (Range 0 to 600 seconds)
                                                  - Default values:  30 sec for eBGP neighbors, 5 sec for iBGP neighbors

 #neighbor {ip|peer-group} timers {keepalive} {holdtime}
                                                  - Changes the default values of BGP timers per specific neighbor or peer group
                                                  - Per neighbor timer overwrites the process timers



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===========================*
   Establishing Peerings
*===========================*
- The command "neighbor 1.2.3.4 remote-as 100" explained:
 > The local router listens for the address 1.2.3.4 starting a TCP session to DST (destination) port 179
      or the local router could initiate a TCP session to 1.2.3.4 on DST port 179.
 > By default the SRC (source) IP is the IP configured on the outgoing interface.
 > This is called the BGP update source and can be manually configured with the "neighbor update-source" command.
 > Recursive lookups are used to determine the outgoing interface to the destination.
 > Unexpected BGP sessions will be refused.
 > The SRC/DST IP addresses, DST port, AS-number and authentication must match between neighbors.
 > If the AS-numbers match between peers, then the session is iBGP according to Cisco IOS, otherwise it is eBGP.
 > Vendor 'J' does not make this distinction.
```

```
- The BGP States are:
 > Idle           - Indicates the router is currently not attempting any connection establishments.
 > Connect        - Indicates the router is waiting for the TCP connection to be completed. If successful an OPEN message is sent.
 > Active         - Indicates the router didn't receive agreement on parameters of establishment and is trying to initiate TCP.
 > OpenSent       - After the TCP session is setup, the router waits for an OPEN message to confirm all parameters.
                  - If no errors a BGP keepalive message is sent.
 > OpenConfirm    - Indicates the router is waiting for a keepalive or notification message.
                  - If a keepalive is received the state changes to Established, else changes to Idle
 > Established    - Indicates peering to a neighbor is established; routing begins.


- The BGP Open Message contains the following fields:
 > BGP version number         - Must match between neighbors.
 > Local AS number            - Must match between neighbors.
 > Holdtime                   - Routers agree on lowest suggested value between neighbors.
 > BGP RID (Router Identifier).
 > Optional parameters.

- Test a connection between peers to confirm connectivity, by using "telnet {peer-ip} 179 /source {int}" .
```

```
-----------
 COMMANDS
-----------
# telnet {peer ip} 179 /source {interface}      - Useful to test connectivity between peers
# sh tcp brief                                  - Lists the TCP sessions, the TCB, the IPs, ports and their states
                                                - Established BGP peers should be in a 'ESTAB' state
# clear tcp tcb {number}                        - Clears a particular tcp session


# debug ip tcp packet detail                    - Good for seeing the TCP session being build, with SRC and DST IPs and ports
# debug ip tcp transactions                     - Shows all TCP transactions (start of session, session errors, etc)
# debug ip bgp events                           - Shows the BGP state transitions
# debug ip bgp keepalives                       - Debugs BGP keepalive packets
# debug ip bgp updates  [acl]                   - Shows all incoming or outgoing BGP updates (!!USE WITH CAUTION!!)
# debug ip bgp [ip] updates [acl]               - Shows all BGP updates received from or sent to a BGP neighbor
                                                - [acl] Optionally matches an IP access-list (recommended)
#router bgp {asn}
 #neighbor {ip|peer-group} remote-as {asn}      - Defines an external/internal neighbor
 #neighbor {ip|peer-group} description {text}   - Assigns a description to an external neighbor. Text can be up to 80 characters
 #neighbor {ip|peer-group} shutdown             - Disables communication with the BGP neighbor
                                                - Recommended while doing extensive modification to routing policies
 #neighbor {ip|peer-group} update-source {int}  - Specifies the source interface for the TCP session that carries BGP traffic
```

Authentication

- BGP only supports MD5 authentication on a per neighbor basis.

COMMANDS

```
#router bgp {asn}
 #neighbor {ip|peer-group} password {pwd}        - Enables MD5 authentication on a specific BGP session
                                                 - {pwd}: Must match on both sides
                                                 - CaSe-SenSiTive, the first character may not be a number
```

eBGP sessions

- The Cisco AD (Administrative Distance) for eBGP peers is 20.
- By default the TTL (time-to-live) is set to 1 for eBGP sessions.
- If an eBGP session is configured between two non-directly connected peers, the TTL must be increased with "ebgp multihop".
- This also applies when a loopback interface is used, as traffic to the loopback counts as one extra hop.

- eBGP loop prevention is done via the AS-path list
  > A router will not accept a prefix if the locally-configured ASN is listed in the received as-path list.
  > This default behavior can be changed with the "neighbor allowas-in" command.

- BGP Backdoor
  > When a router learns a prefix via two paths, one via eBGP and the other via IGP, the eBGP route based on the AD(20) will be chosen as the best.
  > This might not always be the required best route.
  > The AD of that one route could be changed or the BGP backdoor feature could be used, which makes the IGP route the preferred route.

- BGP Maximum-Paths
  > Used to control the maximum number of parallel internal/external BGP routes that can be installed in the routing table.
  > Two required conditions:
  >> All attributes must be the same, i.e. weight, local-pref, as-path, origin and med/IGP distance.
  >> The next-hop router for each multipath must be different.

COMMANDS

```
#router bgp {asn}
 #neighbor {ip} ebgp-multihop [ttl]              - By default, eBGP neighbors must be directly connected (TTL=1)
                                                 - This allows a peer to be several hops away
                                                 - Typically used when eBGP is between loopbacks interfaces
                                                 - If no TTL is entered a command default of 255 is used
 #neighbor {ip} ttl-security hops {hop-count}    - Max number of hops that can separate the eBGP peer from the local router
                                                 - A lightweight security mechanism to protect eBGP sessions from CPU-based attacks

 #neighbor {ip} allowas-in {no}                  - Disables the default eBGP loop-prevention for the specified amount of entries
                                                 - Thereby allowing the local ASN to be listed in a received as-path list
                                                 - {no} The number of times the local ASN may be listed, provided it's on the LEFT
                                                 - If no number is supplied, the default value of 3 is used

 #neighbor {ip} as-override                      - Allows a PE router to override the ASN of a site with the ASN of a provider
                                                 - The "as-override" could be used as an alternative to "allowas-in"

 #distance bgp {ext-ad} {int-ad} {local}         - Sets the AD for eBGP, iBGP, and local routes (default: eBGP=20 & local/iBGP=200)
                                                 - {local}: Locally originated routes via the "network",
                                                       "aggregate" or "redistribution" command

 #network {prefix} backdoor                      - Makes the IGP route more preferred than the eBGP route for the destination
 #maximum-paths eibgp {max-number}               - Control the max number of parallel routes that is allowed to be installed (def=1)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================*
   Next-Hop Processing
*=======================*
- When a packet is passed between iBGP peers, NO next-hop processing is done, unless confederations are used.
- When a packet is passed between eBGP peers, the next-hop field is modified to the IP address of the sending eBGP router's interface
- If the receiving BGP router is in the same subnet as the current next-hop address, the next-hop field remains unchanged
        to optimize packet forwarding (typically seen on multi-access networks).
- Be careful with next-hop processing on NBMA networks. The next-hop must be reachable. Rather use a sub-interface interface
      on a different subnet or alternatively change the next-hop processing.

- Next-hop processing could be changed in one of two ways:
 > As mentioned above with the "neighbor next-hop-self" command.
 > Or with a route-map by setting the "ip next-hop".

-----------
 COMMANDS
-----------
#route-map {name}
 #set ip next-hop {ip}                           - Changes the next-hop to the IP specified

#router bgp {asn}
 #neighbor {ip} route-map {name} {in|out}        - Applies the route-map to change next-hop processing

 #neighbor {ip} next-hop-self                    - Instructs iBGP to use this router as the next-hop for routes advertised
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
    iBGP Sessions
*======================*
- Cisco AD (Administrative Distance) for iBGP peers are 200.
- There is no BGP next-hop modification by default between iBGP neighbors. Thus a full mesh between iBGP neighbors are required
      for full reachability.
- Because iBGP sessions are usually logical, it is recommended to setup iBGP sessions using loopback interfaces
- iBGP loop prevention is done via route suppression/BGP split horizon:
 > I.e. iBGP learned routes will not be advertised onto other iBGP neighbors.
 > This rule implies that the following is inplace:
      >> Fully meshed iBGP peerings (n*(n-1)/2) OR
      >> Route-reflection OR
      >> Confederations

- RRs (Route-Reflectors)
 > Modify the iBGP split-horizon rule.
 > With RRs the actual traffic does not need to pass through the RR, only the updates should.
 > RRs have different client behaviour:
      >> eBGP neighbors
            >>> Normal eBGP neighbors.
            >>> Received updates will be advertised to other eBGP neighbors, RR clients and non-clients.
      >> RR Clients
            >>> Configured with "neighbor route-reflector-client" on the RR.
            >>> Received updates will be advertised to eBGP neighbors, other RR clients and non-clients.
      >> Non-Client peers
            >>> Normal BGP neighbors (non RR clients).
            >>> Received updates will be advertised to eBGP neighbors and RR clients.

 > RR configuration is done only on RRs. RR clients use normal peering configuration, but only to the RR.
 > Always configure a cluster-ID on RRs when they part of a redundant cluster.
 > The default value of the cluster-ID if not configured is the BGP router-ID of the RR (not necessary in non-redundant clusters,
       as the BGP router-ID is already unique).


-  Confederations
 > Autonomous system can be broken up into smaller sub autonomous systems called confederations.
 > Even though confed-peers use eBGP sessions, they exchange routing information as if they were iBGP peers. Specifically,
       the next-hop, Multi_Exit_Discriminator (MED) attribute, and local preference information is preserved.
 > It is generally recommended to use private ASNs (64512-65535).
 > Neighbors inside a confederation AS must still be fully-meshed or route-reflectors must be used.
 > Always start the BGP process with the sub-AS number.
 > Then specify a real AS number.
 > And lastly list the connected sub-AS numbers in the confederation.

CONFIG-SET: BGP Confederations Example
+------------------------------------------------------
|     router bgp 65001                                      - Member-AS number
|      bgp confederation identifier 123                     - Real AS-number
|      bgp confederation peers 65002 65003                  - Confederation peer AS-numbers
|     !
|      neighbor 10.1.1.4 remote-as 65001                    - iBGP neighbor
|     !
|      neighbor 10.1.1.2 remote-as 65002                    - eBGP with intra-confederation AS
|      neighbor 10.1.1.3 remote-as 65003                    - eBGP with intra-confederation AS
|     !
|      neighbor 145.1.1.2 remote-as 102                     - Real eBGP session
|
-----------
 COMMANDS
-----------
#router bgp {asn}                               >>> Route-Reflectors <<<
 #neighbor {ip-address} route-reflector-client  - Configures an iBGP neighbor to be a client of this RR

 #bgp cluster-id {cluster-id}                   - Optionally assigns a cluster-ID to the RR
                                                - Cluster-ID is a 4-byte value
                                                - Required only for clusters with redundant RRs
                                                - Cluster-ID cannot be changed after the first client is configured

                                                >>> Confederations <<<
#router bgp {sub-as-number}                     - Configures BGP process with member-AS number
 #bgp confederation-id {external-as-number}     - Configures real external AS-wide number
 #bgp confederation-peers {list-intra-confed-as}- Defines the connected confederation ASs



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
    iBGP Synchronization
*=========================*
- BGP Synchronization Rule:
 > If an autonomous system is a transit autonomous system and synchronization is enabled, BGP will not advertise a route until
      that router has learned that external route via its IGP.

- This rule was designed to prevent routing blackholes that can arise when non-BGP routers are in the BGP transit path and don't
      have knowledge about the external next-hop destinations. That resulted in traffic being dropped.
- This is a legacy rule, and is disabled  by default as from IOS 12.2(8)T+.

- Alternatives to using synchronization:
 > Run BGP on every router in the transit path.
 > Redistribute BGP into IGP (generally not a good idea in production networks).
 > Tunnel BGP using a tunnel technique like GRE, etc.

-----------
 COMMANDS
-----------
#[no] synchronization                           - Disables synchronization between BGP and an IGP
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*================================*
    BGP Path Attributes
*================================*
- The BGP Path Attributes are categorized as follow
 > Well-known attributes (must be recognized by every BGP implementation)
      >> Mandatory must be present in all updates.
            >>> Next-Hop (see below)
            >>> AS-Path (see below)
            >>> Origin (see below)
      >> Discretionary could be present in an update, but not required.
            >>> Local Preference (see below)
            >>> Atomic Aggregate - Is used to signal that original information may have been lost when the updates were
                  summarized into a single entry.

 > Optional attributes- (not expected to be recognized by all BGP implementations)
      >> Transitive will be propagated if not recognized, but the partial bit will be set to indicate that the attribute
                   was not recognized.
            >>> Aggregator    - Identifies the AS and router within that AS which created the route aggregate.
            >>> Community     - Is used for route tagging (see below).
      >> Non-Transitive will be discarded if not recognized.
            >>> MED (see below).

- Only the best routes (indicated by a '>') are considered candidates for advertising and candidates to be placed in the RIB.
- An outbound routing policy affects inbound traffic.
- An inbound routing policy affects outbound traffic.
- Prerequisites:
 > A prefix must have IGP next-hop reachability for BGP to consider that route.
 > The synchronization rule must be met or disabled.

- The BGP bestpath selection process is based on BGP attributes in the following order:
 1- Prefer the highest Cisco weight (local to router).
 2- Prefer the highest local preference (local to AS).
 3- Prefer the routes that the router originated locally.
 4- Prefer the shortest AS paths (only the length is compared).
 5- Prefer the lowest origin code (IGP-i before EGP-e before Incomplete-?).
 6- Prefer the lowest MED.
 7- Prefer external (eBGP) paths over internal (iBGP) paths:
      >> For eBGP paths, prefer the oldest (most stable) path.
      >> For iBGP paths, prefer the path through the closest IGP neighbor (lowest IGP metric).
 8- If route reflectors are configured:
      >> With multiple iBGP routes, non-reflected routes are preferred above reflected routes.
      >> Then reflected routes with a shorter cluster-list are preferred above routes with a longer cluster-list.
 9- Prefer paths from the router with the lower BGP router-ID.

- Cisco Weight Attribute
 > Used for OUTBOUND routing decisions when ONE router has MULTIPLE links to a provider/providers.
 > It is locally significant within a router as it is never sent out in updates.
 > BGP weights can be specified per neighbor with the "neighbor weight" command or with a "route-map".
 > Weight is applied to new incoming updates to affect OUTBOUND routing decisions.
```

> To enforce newly-set weight values, re-establish BGP sessions with the neighbors (refer to the Clearing BGP Session).
> If no weight value is specified, the default value of 0 is applied to received routes.
> Routes that the router originates locally have a default value of 32768.
> Routes can be matched in any combination of prefix-lists, AS-path filters, or other BGP attributes.
> Routes not matched by the route-map will be discarded.
> Copy descretly owned by Kane Bagwell

- Local Preference Attribute
 > Used for OUTBOUND routing decisions when SINGLE/MULTIPLE routers have SINGLE/MULTIPLE links to a provider/providers.
 > A BGP router can set local preference when processing incoming route updates when doing redistribution, or when sending outgoing route updates.
 > The default value is 100. A higher value is always preferred.

- AS-Path Attribute
 > Used for INBOUND routing decisions to decide which return path to use when MULTIPLE paths exist.
 > AS-path prepending is useful in two scenarios:
 >> Manipulating the outgoing AS-path length to effect proper return path selection for primary/backup links.
 >> Distributing the return traffic load for multi-homed scenarios.
 > To enforce newly-set AS-path lengths, re-send BGP updates outbound to the neighbors (refer to Clearing BGP Session).
 > AS-path prepending should be performed on outgoing eBGP updates over the non-desired return path or the path where the traffic load should be reduced.

!!NOTE!! AS-path prepending cannot be monitored or debugged on the sending router. It can only be observed on the receiving router.

- MED (Multi Exit Discriminator) Attribute
 > Used for INBOUND routing decisions when MULTIPLE return paths from the SAME AS to ONE/MORE routers exist.
 > There is by default no MED attribute attached to a route, except if the router originated the route. The Cisco default MED value of received updates is then assumed to be 0.
 > A received MED value is not propagated outside of the receiving AS.
 > A lower MED value is more preferred.
 > By default, the MED is considered only during selection of routes from the same AS, and does not include intra-confederation autonomous systems.
 > Default MED behavior is different with redistribution. With the "network" or "redistribution" command the metric in the routing table will be used as the MED.

- Community Attribute
 > The community attribute is an optional transitive attribute.
 > BGP communities are a means of tagging routes to ensure consistent filtering or route selection policy in incoming/outgoing routing updates, or with redistribution.
 > By default, communities are stripped in outgoing BGP updates. Sending them must be manually enabled.
 > Routers that do not support communities will pass them along unchanged.
 > There are two types of community attributes:
 >> Standard Communities
 >>> Is an extension to BGP used to pass additional information between BGP peers.
 >>> Typically used to aid policy administration and reduce the complexity of route management.
 >>> Is a 32-bit value.
 >> Extended Communities
 >>> Provides a mechanism for labeling information carried by BGP.
 >>> Is a 64-bit value.

```
 > BGP community formats
      >> Decimal
            >>> The community value is expressed as "set community 1966100".
            >>> By default, IOS uses the older decimal format.
      >> Hexadecimal
            >>> The community value is expressed as "set community 0x1E0014".
      >> ASN:NN
            >>> A new community format is expressed as "set community 30:20"
            >>> The high-order 16-bits contain the ASN of the autonomous system which defines the community meaning.
            >>> The low-order 16-bits have local significance to the originating AS.
            >>> Enabled with the command "ip bgp-community new-format".

 > RFC-1997 define several well-known standard communities:
      >> no-advertise - Do not advertise routes to any BGP peer (local to a router).
      >> no-export    - Do not advertise routes to REAL eBGP peers (local to an AS).
      >> local-as     - Do not advertise routes to any eBGP peers (local to a confed-AS).
      >> internet     - Used to match all communities. (Not RFC standard, but Cisco implementation).

 > Community values specified with the "set" command in a route-map overwrite existing community values unless
      the 'additive' keyword is specified.
```

```
CONFIG-SET: Setting BGP Communities in a Route-Map
+-------------------------------------------------------
|     route-map SET-COMM
|      match ip address 123
|      set community 100:12 100:212 additive              - Attaches the communities to the matching routes
|     !                                                   - [additive] Preserves the original communities and append new ones
|     router bgp 100
|      neighbor 10.5.0.5 route-map SET-COMM out
|      neighbor 10.5.0.5 send-community standard          - By default, communities are not sent in outgoing updates
|
```

```
 > Community-lists are similar to access-lists and are evaluated sequentially, line by line.
 > All the values listed in one line must match for the line to match and permit or deny a route.

 > Standard community-lists
      >> The keyword 'internet' can be used to match any community value.
      >> Permit action = match.
      >> Deny action = don't match.

 > Expanded community-lists
      >> Similar to simple community-lists but allow matching based on regular expressions.
      >> The regular expression '.*' can be used to match any community value.

 > Named community-lists
      >> Allow the use of meaningful names.
      >> Can be configured with regular expressions or with numbered community values.
 > Cost community
      >> Allows the BGP best-path selection process to be customized for a local AS or confederation.
      >> Influences the BGP best-path selection process at the POI (Point of Interest).
      >> Applied ONLY to internal routes by configuring the following:
            #set extcommunity cost [igp] {community-id} {cost-value}

 > BGP dmzlink bandwidth extended community
      >> Used to enable multipath load balancing for external link with unequal bandwidth capacity.
      >> This is done by advertising the bandwidth of a link used to exit the AS.
      >> Supports iBGP and eBGP multipath load balancing.
      >> Indicates the preference of an AS exit link in terms of bandwidth.

-----------
 COMMANDS
-----------
#route-map {name}                               >>> Cisco Weight configs <<<
 #set weight {value}                            - Changes the weight in a route-map
#router bgp {asn}
 #neighbor {ip} weight {weight}                 - Sets the weight for all routes received from the neighbor
                                                - Weight value (1-65535)
 #neighbor {ip} route-map {map-name} in         - Sets the weight for the matching routes from the neighbor


#route-map {name}                               >>> Local-Preference configs <<<
 #set local-pref {value}                        - Changes the local preference in a route-map
#router bgp {asn}
 #bgp default local-preference {pref}           - Changes the default local preference in all updates received from a neighbor
 #neighbor {ip} route-map {map-name} in         - The route-map sets the local-preference to incoming updates from eBGP neighbors
 #neighbor {ip} route-map {map-name} out        - Used to change the local-preference advertised to a iBGP neighbor


#route-map {name}                               >>> AS-Path Prepending configs <<<
 #set as-path prepend as-number [as-number]     - Prepends an ASN to the AS path
#router bgp {asn}
 #neighbor {ip} route-map {map-name} out        - Applies the prepended AS-path to all routes matching
 #bgp bestpath as-path ignore                   - Ignores the AS-path length in its decision process
                                                - Cisco IOS takes the length of the AS-path attribute into consideration but
                                                    RFC 1771 does not include this step


#route-map {name}                               >>> MED configs <<<
 #set metric {value}                            - Changes the MED in a route-map
#router bgp {asn}
 #neighbor {ip} route-map {map-name} in|out     - Applies the new MED value set in the route-map
 #default-metric {number}                       - This changes the default MED value

                                                >>> Advanced MED configs <<<
 #bgp always-compare-med                        - Used to consider MED for routes coming from a different ASs
 #bgp bestpath med missing-med-worst            - Causes a missing MED to be interpreted as infinity (worst)
                                                - If a MED is not attached to a BGP route, Cisco assumes as value 0.
 #bgp bestpath med confederation                - Allows routers to compare MEDs learned from confederation peers
 #bgp deterministic-med                         - Ensures the comparison of MEDs from different neighbors in the same AS
                                                - By default routes from the same autonomous system are grouped,
                                                      and only the best entries of each group is then compared

                                                >>> Community configs <<<
#ip community-list {1-99} {permit|deny} value [value…]
                                                - Defines a standard community-list
#ip community-list {100-199} {permit|deny} {regexp}
                                                - Defines an expanded community-list
#ip community-list {std|exp} {name} {permt|deny} {value | reg-exp}
                                                - Defines a names community-list
#route-map {name}
 #match community {value/list}                  - Reference a community value or list
 #set community {value}                         - Sets a static or well-known community
 #set comm-list {list}                          - Applies the BGP community list
 #set extcommunity cost [igp] {community-id} {value}  - Sets the cost community value

#router bgp {asn}
 #neighbor {ip} route-map {map-name} in|out     - Applies/matches the community values set in the route-map
 #neighbor {ip} send-community [std|ext|both]   - Enables the sending of communities in outgoing updates
                                                - [both] Is the default option if none specified

 #bgp {ip} dmzlink-bw                           - Distributes traffic proportionally over external links,
                                                      with unequal bandwidth when multipath is enabled
 #neighbor {ip} dmzlink-bw                      - Enables the link bandwidth attribute for routes learned to be included
                                                      for the specified neighbor


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===========================*
  Originating Prefixes
*===========================*
- Routes can be originated into BGP in the following ways:
 > Using the "network" statement.
 > By doing redistribution from another protocol.
 > By originating a default route.
 > By using the "aggregate-address" command.
 > By using BGP conditional route injection.

- Network Statement
 > If no mask option is specified a classful subnet will be assumed.
 > If Auto-Summary is ENABLED, at least one subnet of the major network is required in the RIB for a route to be originated in BGP.
 > If Auto-Summary is DISABLED, an exact route match is required in the RIB before the route is originated in BGP.
 > BGP routes originated through the "network" statement have an origin code of 'i' for IGP.
 > Optionally a route-map can be referenced, which allows route parameters to be modified before the route is entered into
      the BGP table.
 > A default route could also be originated using "network 0.0.0.0 0.0.0.0" with same requirements as above.
```

```
- Redistribution
 > Can be used to originate routes into BGP that were learned using another routing protocol.
 > Routes redistributed into BGP will have an origin code of '?' signifying incomplete, since BGP does not know exactly where the
      route was created.

- Default Information Originate
 > The "default-information originate" command is used to configure a BGP routing process to advertise a default route (0/0).
 > A redistribution statement must also be configured to complete this configuration or the default route will not be advertised.
 > The "default-information originate" command, however, requires explicit redistribution of the route 0.0.0.0.

- Neighbor Default Originate
 > When enabled BGP advertises an unconditional default route to the specified neighbor.
 > This advertises the default route to a BGP neighbor even if a default route is not present in the BGP table.
 > The advertisement can be made conditional by using a route-map.
 > The neighbor default route is not passed through the outbound BGP filters (prefix-list, filter-list, or route-map).

- Aggregation
 > This is BGP terminology for summarizing routes in the BGP table.
 > The aggregate will be announced ONLY if at least one of the routes in the specified range is in the BGP table (not the IGP table).
 > The default behavior is to advertise the aggregate and the more specific routes. This can be disabled with 'summary-only' keyword.
 > BGP routes originated through the "aggregate-address" command have an origin code of 'i' for IGP.
 > Optionally, a route-map can be used to suppress only certain routes of the aggregate, using the "suppress-map".
 > Additionally, certain suppressed routes can be unsuppressed on a per-neighbor basis, using the "unsuppress-map".

- See Conditional-Route-Injection and Conditional-Route-Advertisement below for additional methods.

CONFIG-SET: Originating prefixes with BGP
+----------------------------------------------------------------------------------------------------
|      ip prefix-list DEF seq 5 permit 0.0.0.0/0              - Matches the default route
|      !
|      route-map DEFAULT permit 10
|       match ip address prefix-list DEF                      - Reference the prefix-list
|      !
|      route-map COMM
|       set community 100:12                                  - Attaches the communities to the matching routes
|      !
|      router bgp 100
|       no auto-summary                                       - Automatic summarization to the classful boundary is disabled
|       network 10.22.22.0 mask 255.255.255.0                 - The exact prefix 10.22.22.0/24 must be present in the RIB
|       !                                                          table to be originated this into the BGP table
|       network 10.1.1.0 mask 255.255.255.0 route-map COMM    - Community 100:12 will be attached to this route when originated
|       !
|       aggregate-add10.22.0.0 mask 255.255.0.0 summary-only  - The summary 10.22.0.0/16 will only be originated if a more
|       !                                                          specific route of the summary is present in the BGP table
|       neighbor 10.1.34.4 remote-as 65000
|       neighbor 10.1.34.4 default-orig route-map DEFAULT     - Because of the route-map, a default route will only be sent to
|       !                                                          10.1.34.4 if a default route exists in the BGP table
|       neighbor 10.1.13.1 remote-as 65001
|       neighbor 10.1.13.1 default-originate                  - A default route will be sent to 10.1.13.1 regardless
|      !                                                           whether a default exists in the BGP table
```

COMMANDS

```
#router bgp {asn}
 #network {network} [mask {net}] [route-map {name}]   - Originates a network into BGP with an origin code of IGP (i)
 #redistribute {igp} [pid] [metric] [route-map]       - Originates redistributed routes with an origin code of '?-incomplete'
 #redistribute {static|conn} [route-map] [metric]     - Originates static or connected routes into BGP

 #default-information originate                 - Originates a redistributed default route
 #neighbor {ip} default-originate [route-map]   - Advertises an unconditional default route to the neighbor
                                                - [route-map] Allows the advertisement to be conditional
 #default-metric {metric}                       - Set the metric of redistributed routes if not specified
 #aggregate-address {aggregate} [mask] [summary-only]
                                                - Originates a BGP summary/aggregate address
                                                - [summary-only] Advertises only the aggregate and not the individual routes

 #aggregate-address {aggregate} {mask} suppress-map {route-map}
                                                - Specifies only a subset of routes to be suppressed
                                                - The routes within the aggregate permitted/matched by the route-map will be
                                                    suppressed from being advertised to neighbors

 #neighbor {ip} unsuppress-map {route-map}      - Specifies what aggregate routes to unsuppress on a per neighbor basis
```

-=-

===============

Filtering

===============

- The aggregate command could be used to filter the advertisement of BGP routes (refer to previous Section).

- AS-Path Filters
  - > Can be used to selectively filter routes based on the AS-path list.
  - > Incoming routes- Permitted routes are entered into the local BGP table, denied routes are silently dropped.
  - > Outgoing routes- Permitted routes are transmitted to the neighbor, denied routes are never sent to the neighbor.
  - > Refer to the Regular Expression Section below to understand REGEX better.

- Distribute-Lists
  - > Allow BGP routes to be filtered per neighbor using an ACL.
  - > Standard ACLs have an inherent problem- they cannot match the length of a network mask exactly, thus allowing more than the intended length. The following example ACL will permit the /19 aggregate as well as the more specific /24 networks.

```
        #access-list 1 permit 10.10.0.0 0.0.31.255
```

  - > Extended ACLs can be used to match the network mask exactly. The following ACL line achieves the required match of 10.10.0.0/19:

```
        #access-list 101 permit ip 10.10.0.0 0.0.0.0 255.255.224.0 0.0.0.0.
```

    - >> The source is 10.10.0.0 and the source-wildcard of 0.0.0.0 is configured for an exact match of source.
    - >> A mask of 255.255.224.0, and a mask-wildcard of 0.0.0.0 is configured for an exact match of source mask.

> For more information on this use of ACLs refer to the Security Chapter.
> Generally, when filtering BGP networks, prefix-lists would be used. Occasionally when odd and even networks should be filtered while controlling the mask length, a distribute-list with an extended ACL could still be used.

CONFIG-SET: BGP Distribute-List Example

```
+-----------------------------------------
|     access-list 101 permit ip 10.10.0.0 0.0.0.0 255.255.224.0 0.0.0.0
|     access-list 123 permit ip 10.20.0.0 0.0.254.0 255.255.255.0 0.0.0.255
|     !
|     router bgp 100
|      neighbor 172.16.1.2 remote-as 200
|      neighbor 172.16.1.2 distribute-list 101 in           - Filters 10.10.0.0/19 from 172.17.1.2
|      neighbor 172.20.1.1 remote-as 200
|      neighbor 172.20.1.1 distribute-list 123 in           - Filters routes between /24 and /32 with an even number
|                                                                in the 3rd octet
```

- Prefix Filter
 > A prefix-list filter can be used as an alternative to ACLs to filter BGP routes.
 > Prefix lists are a more convenient way to filter networks in BGP.

CONFIG-SET: BGP Prefix-List Examples

```
+-------------------------------------------------------
|     ip prefix-list A permit 0.0.0.0/0 ge 32        - Matches all hosts routes
|     ip prefix-list B permit 0.0.0.0/1 ge 9         - Any subnets in Class A address space (/1: 1st bit(0) can't change)
|     ip prefix-list C permit 128.0.0.0/2 ge 17      - Any subnets in Class B address space (/2: 1st 2 bits(10) can't change)
|     ip prefix-list D permit 192.0.0.0/3 ge 24      - Any subnets in Class C address space (/3: 1st 3 bits(110) can't change)
|     ip prefix-list E permit 0.0.0.0/0 le 32        - Match any/all routes
|     ip prefix-list F permit 0.0.0.0/0              - Match just the default route
|     ip prefix-list G permit 0.0.0.0/1 le 24        - Matches any prefix in Class A address space with more than 256 addresses
|     ip prefix-list H permit 10.0.0.0/8             - Matches only a 10.0.0.0/8 route (no more, no less)
|     ip prefix-list I permit 10.0.0.0/8 le 32       - Matches any route in the RFC-1918 pvt 10/8 range (including 10.1.2.0/24)
|     ip prefix-list J permit 172.16.0.0/12 le 32    - Matches any route in the RFC-1918 pvt 172.16/12 range
|     ip prefix-list K permit 192.168.0.0/16 le 32   - Matches any route in the RFC-1918 pvt 192.168.0.0/16 range
```

 COMMANDS

```
# sh ip as-path-access-list [filter-list]        - Shows the configured filter lists
# sh ip bgp filter-list {access-list-number}     - Shows all routes permitted by the specified AS-path access-list
# sh ip bgp regexp {expression}                  - Shows all routes matching regular-expression in one or all filter-lists
# sh ip prefix-list {list}[det|sum][longer]      - Shows the prefix-list and the sequence numbers
# sh ip bgp prefix-list {list-name}              - Shows all routes in the BGP table matching prefix-list

#ip as-path access-list {1-199} [permit/deny] {regex}
                                                 - Configures an AS-path filter list
#ip prefix-list {name} [seq] [permit|deny] {prefix} [ge] [le]
                                                 - Configures a prefix-list, if [ge/le] is not defined, prefix is matched exactly
                                                 - [ge] Means greater than AND equals to
                                                 - [le] Means less than AND equals to
```

```
#router bgp {asn}
 #neighbor {ip} filter-list {as-path} [in|out]  - Applies an AS-path filter list to the neighbor
 #neighbor {ip} prefix-list {list} [in|out]     - Applies a prefix-list filter to the neighbor
 #neighbor {ip} distribute-list {acl} [in|out]  - Applies a ACL distribute-list to the neighbor
 #distribute-list {acl} {in|out} [interface]    - Filters redistributes routes using an ACL distribute-list
 #distribute-list prefix-list {name} (in|out} [interface]
                                                - Filters redistributes routes using a prefix distribute-list



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
    Regular Expressions
*============================*
- A regular expression is a pattern used to match against an input string.
- When a regular expression is parsed, the input must match the string specified.

- A regular expression comprises of:
 > A Range, which is a sequence of characters:
      [ ]   (Square Brackets) - Represents a range of characters.

 > An Atom, which is single characters:
      |     (Vertical Bar)     - Represents 'OR' statements.
      \     (Backslash)        - Matches an exact character. Removes special meaning if any.
      ( )   (Parenthesis)      - Represents 'and' operations. Used to group things together.
      .     (Dot)              - Matches any single character.
      ^     (Carrot)           - Matches beginning of an input string.
      $     (Dollar)           - Matches the end of an input string.
      _     (Underscore)       - Matches any delimiter (beginning, end, space, tab, comma).

 > A Piece, which is a single character that applies repetition to the Atom/Character that immediately precedes it:
      *     (Asterisk)         - Matches ZERO or MORE Atoms.
      ?     (Question Mark)    - Matches ZERO or ONE Atoms.
      +     (Plus)             - Matches ONE or more Atoms.

 > A Branch, which is 0 or more concatenated pieces.

- Examples using simple regular expressions:
      213|310                  - Matches 213 or 310.
      [1-4]                    - Matches any number between 1 and 4.
      [67]                     - Matches either 6 or 7.
      [1-4].[67]               - Matches 1/2/3/4, then anything character, then 6/7, i.e. '136' or '417'.
      ^21                      - Matches 21 only at the beginning of the line.
      $31                      - Matches 31 only at the end of the line.
      _41_                     - Matches 41 in the beginning, middle or end of the line.
      (213|218)_31             - Matches 213 or 218 followed by 31, i.e. '213 317' or '218 31'.
      _23(_78)*_45_            - Matches "23 45"  or  "23 78 45"  OR  "23 78 78 78 78 45".
      _23(_78)?_45_            - Matches "23 45"  OR  "23 78 45".
      _23(_78)+_45_            - Matches "23 78 45"  OR  "23 78 78 78 78 78 78 45".
      ^\(213_                  - Matches (213 at the beginning of string.
```

- In the case of BGP, the string specified consists of path information that an input must match.
- Examples using regular expression to match BGP AS-paths:

```
      _100_                    - Passes/passed through AS 100.
      ^100$                    - Directly connected to AS 100 (begins and ends in AS 100).
      _100$                    - Originated in AS 100.
      ^100_                    - Matches networks behind AS 100.
      ^[0-9]+$                 - Matches any AS path that is one AS long.
      ^([0-9]+)(_\1)*$         - Networks originating in neighboring AS, with possible prependings.
      ^$                       - Networks originating in LOCAL AS.
      .*                       - Matches everything.
```

-=-

======================================

BGP Conditional Route Advertisement

======================================

- DOC-CD LOCATION
  - > 12.4T Configuration Guides
  - > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
  - > Configuring a Basic BGP Network
  - > Aggregating Route Prefixes Using BGP

- Assume the following topology:

```
     AS#100-------AS#200
           \      /
            \    /
             \  /
          AS#300
             |
          Prefix-A
```

- AS-300 wants ALL traffic to prefix-A to enter from AS-200 only, but in the event of link failure between AS-200 and AS-300, traffic should be allowed to enter from AS-100.
- AS-100 has a weight set, preferring its direct link to AS-300 for all prefixes.
- Assuming AS-100 is not co-operating in removing the weight value set, what can be done to meet the criteria?

- BGP conditional route advertisement offers an alternative way to affect how traffic enters an AS.
- By conditionally not advertising prefix-A to AS-100, AS-100 is forced to route via AS-200.
- Then, in the event of a link failure, conditional advertisement will advertise prefix-A to AS-100.

- By controlling which prefixes get advertised to which neighbors, traffic can be forced to enter on the appropriate links.
- BGP conditional route advertisement consists of two parts:
  > The prefix/s to watch (LINK-300-200).
  > The prefix/s to advertise (PREFIX-A).

!!NOTE!! Confirm the above prefixes are in the BGP table before configuring conditional route advertisement.

- Once the prefix (LINK-300-200) leaves the BGP table, the prefix (PREFIX-A) will be advertised to AS-100 (100.1.1.1)

```
CONFIG-SET: BGP Conditional Route Advertisement
+-------------------------------------------------------
|     ip prefix-list PREFIX-A permit 30.0.0.0/24                 - Matches the advertised prefix
|     ip prefix-list LINK-300-200 permit 30.20.1.0/30            - Matches the watched prefix
|     !
|     route-map ADV permit
|      match ip address prefix-list PREFIX-A                     - References the advertised prefix
|     !
|     route-map WATCH permit
|      match ip address prefix-list LINK-300-200                 - References the watched prefix
|     !
|     router bgp 300
|      neighbor 100.1.1.1 remote-as 100
|      neighbor 100.1.1.1 advertise-map ADV non-exist-map WATCH   - Applies conditional route advertisement for AS-100
|
>     #sh ip bgp neighbors 100.1.1.1 | i Condition               - A positive, the WATCH route is DOWN
>           Condition-map WATCH, Advertise-map ADV, status: Advertise    (prefix is advertised)
>     #sh ip bgp neighbors 100.1.1.1 | i Condition               - A negative, the WATCH route is UP
>           Condition-map WATCH, Advertise-map ADV, status: Withdraw     (prefix is not advertised)
>
```

```
-----------
 COMMANDS
-----------
# sh ip bgp neighbors {ip}| i Condition          - Shows the condition status of the advertised route

#router bgp {asn}
 #neighbor {ip} advertise-map {route-map} non-exist-map {route-map}
                                                     - Conditionally advertises a route to neighbors based on the existence of another
                                                     - {adv-map}: This is the route to be advertised based on
                                                     - {non-exist-map}: Routes that will be tracked
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
```

BGP Conditional Route Injection

```
*====================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
         > Configuring a Basic BGP Network
          > Originating BGP Routes
```

- Provides a method to originate a prefix into the BGP routing table without the corresponding match in the IGP table.
- Only prefixes that are equal to or more specific than the original prefix may be injected.
- This is used to improve the accuracy of route aggregation, by conditionally injecting or replacing less specific prefixes with more specific prefixes.

```
CONFIG-SET: BGP Conditional Route Injection
+-----------------------------------------------------
|     ip prefix-list ROUTE permit 10.1.1.0/24                     - The route to be monitored
|     !
|     ip prefix-list ROUTE_SOURCE permit 10.2.1.1/32              - The advertising source
|     !
|     ip prefix-list ORIGINATE_ROUTES permit 10.1.1.0/25          - The more specific routes to be injected
|     ip prefix-list ORIGINATE_ROUTES permit 10.1.1.128/25        - The more specific routes to be injected
|     !
|     route-map LEARNED_PATH permit 10
|      match ip address prefix-list ROUTE                         - Watches the monitored prefix in the RIB
|      match ip route-source prefix-list ROUTE_SOURCE             - Matches the prefix learned from a specific source
|     !
|     route-map ORIGINATE permit 10
|      set ip address prefix-list ORIGINATE_ROUTES                - Identifies the specifics to inject
|      set community 14616:555 additive                           - Sets optional parameters
|     !
|     router bgp 3741
|      bgp inject-map ORIGINATE exist-map LEARNED_PATH            - Applies conditional route injection
|
```

```
-----------
 COMMANDS
-----------
#router bgp {asn}
 #bgp inject-map {map} exist-map {map} [copy-attribute]
                              - inject-map      : Defines the prefixes that will be created and installed into the local BGP table
                              - exist-map       : Specifies the prefix which the BGP speaker will track
                              - copy-attr       : Config the injected route to inherit the attributes from the tracked route
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
  Clearing BGP Sessions
*========================*
- The Cisco IOS software command summary lists the following circumstances when a BGP connection should be reset:
 > Additions or changes to BGP-related access lists.
 > Changes to BGP-related weights/attributes.
 > Changes to BGP-related distribution-lists.
 > Changes to BGP-related timers.
 > Changes to the BGP administrative distance.
 > Changes to BGP-related route-maps.

- Traditional Clearing of BGP Session (aka Hard Reset)
 > Completely tears down the BGP session and rebuilds the sessions (interruptive in production).
 > A new session should be re-established within 30-60 sec depending on the amount of routes.
 > If dampening is enabled a hard reset will result in a penalty.
 > Processing the full internet table after a hard reset can take a long time, so be careful with this in production.
```

```
- Soft Reconfiguration: Outbound or Inbound (IOS 11.2+)
 > Outbound soft reconfiguration resends the complete BGP table. It is not configurable and is always enabled.
 > Inbound soft reconfiguration stores a complete copy of the BGP table from a neighbor in router memory (possibly
     resource demanding).
 > Inbound soft reconfiguration must be enabled if needed.

- Route Refresh (Soft Reset) (IOS 12.1+)
 > Used to request a neighbor to resend routing info. Useful after config changes to update the BGP table.
 > Route-refresh-capability is negotiated upon BGP peer session establishment.
 > Is also used with ORF when inbound prefix-list route refresh is required (see ORF Section).

- BGP Dynamic Update Peer-Group Feature
 > Used to recalculate all BGP update-group member sessions.

-----------
 COMMANDS
-----------

                                             >>> Hard-Reset <<<
# clear ip bgp {*|ip|peer-group name}        - Tears the BGP sessions down completely and re-establishes them again


#router bgp {asn}                            >>> Soft Reconfiguration <<<
 #neighbor {ip} soft-reconfig [inbound]      - Enables inbound soft reconfiguration on the router, so that all
                                                   the routes are stored in memory before filters are applied.

# clear ip bgp {ip} soft in                  - This takes all the routes in memory, reapplies the filters, before
                                                   implementing the passed routes into BGP table.
# clear ip bgp {ip} soft out                 - This will resend the BGP table to a neighbor, for that neighbor to re-apply
                                                   all his configured inbound filters


                                             >>> Route Refresh <<<
# clear ip bgp {*|ip|peer-group name} in     - Requests a neighbor to resend routing information without terminating the session

# clear ip bgp update-group [index-group][peer] - Used to recalculate all BGP update-group member sessions


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
  ORF (Outbound Route Filtering)
*==================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
         > Connecting to a Service Provider Using External BGP
          > Influencing Outbound Path Selection


- The purpose of outbound route filtering is to reduce the amount of BGP traffic and CPU use needed to process routing updates.
- With ORF routers exchange inbound filter configurations, which are then used as outbound filters on neighboring routers.
```

- ORF entries are part of the route refresh message.
- Negotiation of prefix-list ORF capability is done during the BGP session setup.
 > The side that has the prefix-list uses the 'send' option, and is configured with the prefix-list inbound.
 > The side that sends the routes uses the 'receive' option.
 > ORF requires the session to be reset after configuration.

- Inbound route refresh is required, and only the inbound prefix-list filter is pushed to the neighbor and used by that neighbor the outbound direction.
- An ORF-capable BGP speaker will install ORFs per neighbor.

```
-----------
 COMMANDS
-----------
# sh ip bgp neighbor                            - Useful to verify neighbor capabilities
# clear ip bgp {ip} in [prefix-filter]          - Triggers a route refresh from ORF receivers
                                                - [prefix-filter] Refreshes the remote filter

#router bgp {asn}
 #neighbor {ip} capability orf prefix-list {send|receive|both} - Enables negotiation of prefix-list ORF capability
 #neighbor {ip} prefix-list {name} in           - Specifies the prefix that will be send to the ORF capable neighbor
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
    BGP Network Migration
*==========================*
```

- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
         > Configuring Advanced BGP Features
          > BGP Network Autonomous System Migration

```
-----------
 COMMANDS
-----------
#router bgp {asn}
 #neighbor {ip} local-as {asn} [no-prepend [replace-as] [dual-as]]
                                                - Hide local-AS feature is necessary to connecting to different ISPs with
                                                        more than one ASN number
                                                - [no-prepend] Does not prepend the "local" ASN to any routes received
                                                - [replace-as] Prepends only the "local" ASN in the AS-path
                                                       The configured ASN from the BGP process is not prepended
                                                - [dual-as] Configures the eBGP neighbor to establish peering
                                                       session with either real ASN or both

 #neighbor {ip} remove-private-as               - Private AS numbers are only removed from the tail-end (left-side)
                                                       of the AS-path before the update is sent
                                                - Private AS numbers followed by a public AS number are not removed
```

BGP Route-Maps

- Default statement is "permit".
- Default sequence number is 10 and the default increment is 5.
- If route is not matched by any statements it is dropped.
- 'Permit all' is achieved by specifying a "permit" without a "match" clause.
- Match conditions in one statement are AND'd together.

CONFIG-SET : BGP Route-Map Example Filtering Routes

```
+---------------------------------------------------------
|     route-map RMAP permit 45
|      match ip address prefix-list LIST              - Allows only matched routes
|     !
|     router bgp 1
|      neighbor 10.1.1.1 route-map RMAP in            - Prefixes not permitted by the route-map are discarded
|
```

- MATCH Criteria:
  > Network number and subnet matched with an IP-prefix list
  > Route originator
  > BGP next-hop address
  > BGP origin
  > Tag attached to IGP route
  > AS-path
  > BGP community attached to BGP route
  > IGP route type (internal/external)

- SET Options:
  > Origin
  > BGP community
  > BGP next-hop
  > Local preference
  > Weight
  > MED

- Route-Map Policy-List
  > Adds the capability for a network engineer to group route-map match clauses into named lists called policy-lists.
  > Policy lists with groups of match clauses can be pre-configured and then referenced within different route maps.
  > Eliminates the need to manually reconfigure each recurring group of match clauses that occur in different route-maps.

- Route-Map Continue Feature
  > Introduces the 'continue' clause to BGP route-map configuration, providing more programmable policy configuration and route filtering.
  > Configures a route-map to go to another route-map entry with a higher sequence number.
  > The 'continue' clause will be executed in the route-map instance if a match occurred..

```
CONFIG-SET: Route-Map Continue Feature
+-----------------------------------------
|     route-map MYNAME permit 10
|      match ip add 1
|      set as-path prepend 2001
|      continue 30                                - Traffic matching sequence-10 will continue to sequence-30
|     !
|     route-map MYNAME permit 20
|      match next-hop 10.1.2.3
|      set local pref 150
|     !
|     route-map MYNAME permit 30
|      set as-path prepend 2001 2001

-----------
 COMMANDS
-----------
# sh route-map [name]                             - Shows the configured route-map/s
# sh ip bgp route-map {name}                      - Executes the route-map against the BGP routing table entries
# sh ip policy-list {name22}                      - Shows the policy list/s

#ip policy-list {name22} {permit | deny}          - Creates a policy list
#route-map {name} [permit|deny] {seq_no}          - Configures the route-map
  #match policy-list {name22}                     - Matches a specific criteria like a policy-list
  #set {parameter}                                - Executes various set functions
  #continue {seq}                                 - Moves on to the specified sequence number


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
  BGP Route-Dampening
*======================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
         > Configuring Advanced BGP Features
          > BGP Route Dampening
```

- Designed to reduce router processing load caused by unstable routes.
- Each time an eBGP route flaps, it gets 1000 penalty points (this cannot be configured or changed).
- iBGP routes are not dampened.
- The penalty placed on a route decays according to the exponential decay algorithm.
- When the penalty exceeds the suppress limit, the route is dampened (no longer used or propagated to other neighbors).
- A dampened route is propagated again when the penalty drops below the reuse limit.
- A route is never dampened for more time than the maximum suppress limit.
- An unreachable route with a flap history is put in the history state. It stays in the BGP table but only to maintain the flap history (marked with 'h' in the BGP table).
- A penalty is applied on the individual path in the BGP table, not on the IP prefix.
- Using "clear ip bgp *" is regarded as a flap to neighbors, which could cause that path to be suppressed.
- Using "clear ip bgp * [soft] in" is NOT regarded as a flap to neighbors.

```
-----------
 COMMANDS
-----------
# sh ip bgp dampened-paths                      - Shows the dampened routes
# sh ip bgp flap-stat [regexp|filter-list|ip]   - Shows flap statistics for all routes with dampening history

# clear ip bgp {ip} flap-stat [regexp|filter-list|prefix]
                                                - Clears the flap statistics but does not release dampened routes
# clear ip bgp dampening [prefix]               - Releases all the dampened routes or just the specified network
                                                - Flap statistics also cleared when the BGP session with the neighbor is lost
# debug ip bgp dampening                        - Shows the BGP dampening events

#route-map name                                 - Route-map to configure dampening for specifics routes only
 #match ip addess {acl}
 #set dampening [half-life][reuse][suppress][max-suppress-time]

#bgp dampening [half-life][reuse][suppress][max-suppress-time] [route-map name]
                  [half-life]                   - Decay time in which the penalty is halved (Def = 15min)
                  [suppress]                    - The value at which a route is dampened (Def = 2000)
                  [reuse]                       - The value when the dampened route is reused (Def = 750)
                  [max-suppress-time]           - Maximum time to suppress the route (Def = 60Min)
                  [route-map]                   - Using route-map to dampen specific routes
                                                - Specified without a route-map applies to all routes


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================*
  Peer-Groups
*==================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
         > Configuring a Basic BGP Network
          > Peer-Groups

- Benefits of using Peer-Groups
 > Reduce the amount of system resources (CPU and memory) necessary in the update generation.
 > Mostly used to simplify large repeating BGP configurations.

- Individual parameters specified in a peer group can be overridden or removed on a neighbor-by-neighbor basis.
- Configurable parameters include the following:
  > Community propagation.
  > Source interface for TCP session.
  > eBGP multihop sessions.
  > MD5 password.
  > Neighbor weight.
  > Filter-lists and distribute-lists.
  > Route-maps.
```

```
----------
 COMMANDS
----------
# sh ip bgp peer-group [peer-group-name]        - Shows the specified peer group or all peer groups
# sh ip bgp peer-group [peer-group-name] summary- Shows summary status of all neighbors in the peer group
# clear ip bgp [peer-group-name] [[soft] in|out]- Clears BGP session with all peer group members
# debug ip bgp groups [index-group] [peer-ip]   - Shows info about peer-group update-group calculation,
                                                       the additions and the removals of members
                                                - Shows info about peer groups, peer-policy, and peer-session templates
#router bgp 1
 #neighbor {group-name} peer-group              - Creates a BGP peer group
                                                - Peer group names are case-sensitive
 #neighbor {group-name} {any-bgp-parameter}     - Specifies any BGP parameter for the peer group
 #neighbor {ip} peer-group {group-name}         - Assigns a BGP neighbor to a peer group, thus inheriting the peer-group parameters
 #neighbor {ip} {any-bgp-parameter}             - Overrides the BGP parameter specified for the peer group
 #no neighbor {ip} {any-bgp-parameter}          - Removes the BGP parameter specified for the peer group


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================*
   Peering Templates
*====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
         > Configuring a Basic BGP Network
          > Peer Templates

- There are two types of peer templates:
 > Peer Session Templates- Used to group and apply the configuration of general session commands to groups of neighbors that
      share common session configuration elements.
 > Peer Policy Templates- Used to group and apply the configuration of commands that are applied within specific NLRI
      configuration mode.

CONFIG-SET: BGP Peer-Templates
+----------------------------------------
|     router bgp 100
|      template peer-policy POLICY                         - Creates a peer policy template, enter policy-template config-mode
|       route-reflector-client                             - Specifies the client as an RR-client
|       weight 300                                         - Specifies a weight for all routes from a neighbor
|     !
|      template peer-session iBGP                          - Creates a peer session template, enter session-template config-mode
|       remote-as 100                                      - Configures peering ASN with a remote neighbor
|       update-source Loopback1                            - Use the Loopback interface for sourcing traffic
|     !
|      neighbor 7.7.2.2 inherit peer-session iBGP          - Sends a peer session template to a neighbor to inherit
|      neighbor 7.7.2.2 inherit peer-policy POLICY         - Configures this peer session template to inherit the configuration
|      neighbor 7.7.4.4 inherit peer-session iBGP          - Sends a peer session template to a neighbor to inherit
|      neighbor 7.7.4.4 inherit peer-policy POLICY         - Configures this peer session template to inherit the configuration
```

Fast External Fallover

- Fast external fallover for external peers is triggered by a session flap, based upon the receipt of an interface change notification.
- By default, when a local BGP interfaces goes down, the BGP neighbors on that interface are shutdown as soon as an interface reset is detected, instead of waiting for the holddown timer (default = 180 sec) to expire.
- If BGP fast external fallover is disbled BGP will wait for the holddown timer to expire before shutting down the neighbor sessions.

COMMANDS

```
#router bgp {asn}                              >>> Global Configuration <<<
 #no bgp fast-external-fallover                - Disables fast external fallover globally. Will wait for hold-time to expire

#interface s0/0                                >>> Interface Configuration <<<
 #ip bgp fast-external-fallover permit         - Allows per-interface fast external fallover
 #ip bgp fast-external-fallover deny           - Prevents per-interface fast external fallover
 #no ip bgp fast-external-fallover             - ONLY removes previously configured interface config, does not disable fall-over
```

BGP Fast Peering Session Deactivation

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
    > Configuring BGP Neighbor Session Options
     > BGP Fast Peering Session Deactivation

- Fast peering enables BGP to monitor the peering session of a specified neighbor for adjacency changes to deactivate that peering session.
- BGP fast peering session deactivation is event driven and is configured on a per-neighbor basis.
- Adjacency changes are detected and terminated peering sessions are deactivated inbetween the BGP scanning intervals.
- A route-map can be used to deactivate the peering session based a specific prefix.
- Only the "match ip address" and "match source-protocol" commands are supported in fast peering route-maps.

CONFIG-SET: BGP fast peering session fall-over

```
+-----------------------------------------------------
|     ip prefix-list FILTER28 seq 5 permit 0.0.0.0/0 ge 28      - Matches any route with a prefix of /28 or more
|     !                                                              specific prefixes
|     route-map CHECK-NBR permit 10
|      match ip address prefix-list FILTER28                    - References the filter
|     !
|     router bgp 45000
|      neighbor 192.168.1.2 remote-as 40000
|      neighbor 192.168.1.2 fall-over route-map CHECK-NBR       - Resets the session if a /28 or more specific
|                                                                    prefix disappears
```

COMMANDS

```
#router bgp {asn}
 #neighbor {ip} fall-over [bfd | route-map]     - Enables BGP fast peering session fall-over
```

Support for Next-Hop Address Tracking

- This is enabled by default when a supporting Cisco IOS software image is installed.
- BGP prefixes are automatically tracked as peering sessions are established.
- Next-hop changes are rapidly reported to the BGP routing process as they are updated in the Routing Information Base (RIB).
- This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a best-path calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

COMMANDS

```
#router bgp {asn}
 #no bgp nexthop trigger enable                 - Disables next-hop tracking (enabled by default)
```

Maximum-Prefix

- DOC-CD LOCATION
  - > 12.4T Configuration Guides
    - > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
      - > Configuring BGP Neighbor Session Options
        - > BGP Neighbor Session Restart After the Max-Prefix Limit Is Reached

```
#neighbor {ip} maximum-prefix {max no} [threshold] [warning-only] [restart {interval}]
                                                    - Controls how many prefixes can be received from a neighbor
                                                    - [Threshold]: The percentage when message is logged (default is 75%)
                                                    - [Warning-only]: When exceeding the maximum number apose to dropping the session
                                                    - [Restart] : Re-establish the session after the specified interval in minutes
```

BGP PA (Policy Accounting)

- BGP PA measures and classifies IP traffic that is sent to or received from different peers.
- PA is enabled on an input interface and counters based on parameters such as community-lists, ASN and AS-paths are used to identify the IP traffic.

```
----------
 COMMANDS
----------
#router bgp {asn}
 #bgp-policy {accounting|ip-prec-map}           - PA is based on community-lists, ASN, AS-paths
                                                - IP-prec-map: QOS policy based on the IP precedence
  #set traffic-index {bucket-number}            - Range (1-8) representing the bucket into which packet and byte statistics
                                                       are collected for a specific classification
 #table-map {name-of-route-map}                 - Enables BGP policy accounting
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
    Troubleshooting BGP
*==========================*
---------------------------------------
   BGP Session Start-Up Problems
---------------------------------------
- Are you seeing the expected neighbor in a NON 'idle' or 'active' state?           # sh ip bgp summary
- Is a sourced telnet to the neighbor address working?                              # telnet {peer-ip} 179 /source {src-int-ip}
- Confirm if the config is correct and matching the neighbor's configuration?       # sh run | b router bgp

- If eBGP, is the neighbor directly connected (Should be one hop in the trace)?     # trace {peer-ip} source {src-int-ip}
 > If not directly connected is multi-hop configured?                               # sh run | i {peer-ip}.*ebgp-multihop

- Is there IP reachability to the neighbor?                                         # ping {peer-ip} source {src-int-ip}
- Is the underlying routing in place between neighbors?                             # sh ip route {peer-ip}

- If the obvious checks don't help, enable debugging to analyze the session setup   # debug ip tcp transactions or # sh tcp brief
 > If the TCP-SYN packet is not answered with a SYN-ACK packet and times out?
      >> Look for ACLs blocking port TCP-179.                                       # sh ip interface | i line|list

 > If the TCP-SYN packet is answered with a RST packet, it verifies reachability,
     but the neighbor is not willing to grant the connection attempt.
      >> Does the neighbor have BGP configured or BGP "neighbor shutdown"?          # sh run | i {peer-ip}.*shutdown
      >> Does the outgoing interface IP match the peer's "neighbor" statement?      # sh run | i neighbor.*{peer-ip}
      >> If not, is the correct source interface specified?                         # sh run | i {peer-ip}.*update-source

 > If the 3-way TCP handshake completes but the router drops the session shortly after causing
     the neighbor to oscillate between idle and active, check the BGP parameters.
      >> Confirm that the AS numbers between the neighbors are correct.             # sh run | i router bgp|remote-as
      >> If using confederations, double check the AS numbers.                      # sh run | i router bgp|remote-as
      >> Is MD5 password authentication configured correctly?                       # sh run | i neighbor.*password

 > Are any TCP session stuck in the TCP handshake?                                  # sh tcp brief
      >> Clear the TCP session.                                                     # clear tcp tcb {number}
```

```
----------------------------------
   Route Selection Issues
----------------------------------
- Are locally originated routes appearing in the BGP table?                        # sh ip bgp
 > If auto-summary is enabled, is at least one subnet of the major network         # sh run | i router bgp|summary
      present in the RIB?                                                          # sh ip route {prefix} longer-prefixes
 > If auto-summary is disabled, is there an exact prefix match in the RIB?         # sh ip route | i {prefix}/{mask}
 > Is a distribute-list configured blocking the prefixes?                          # sh run | i distribute-list

- Is an aggregate is configured but not advertised?                                # sh run | i aggregate
 > Is there a more specific prefix of the aggregate in the BGP table?              # sh ip bgp {prefix}/{mask} longer-prefixes

- Is a prefix in the BGP table not getting advertised to a iBGP neighbor?          # sh ip bgp {prefix}     (YIELDS NO RESULT)
 > Was the prefix learned via iBGP? BGP split horizon (Look for 'i' routes)?       # sh ip bgp {prefix} | i _i|>i

- Are you receiving any prefixes from the neighbor (look at 'PfxRcd')?             # sh ip bgp summary | i {peer-ip}
      >> Is the neighbor sending any routes (this done on neighbor)?               # sh ip bgp neighbor {peer-ip} advertised-route
 > Are the prefixes showing BEFORE any filters are applied (need "soft-reconfig")?  # sh ip bgp neighbor {peer-ip} received-routes
 > Are the prefixes showing AFTER the filters were applied?                        # sh ip bgp neighbor {peer-ip} routes
      >> If not, are any prefix-filters configured denying the prefixes?           # sh run | i {peer-ip}.*prefix-list
      >> If not, are any AS-path filters configured denying the prefixes?          # sh run | i {peer-ip}.*filter-list
      >> If a route-map is configured:                                             # sh run | i {peer-ip}.*route-map
            >>> The routes must be explicitly permitted to be accepted/used.       # sh route-map {name}
            >>> Are the prefixes explicitly denied?                                # sh route-map {name}
 > Was the BGP session cleared after changes to filters and route-maps?            # clear ip bgp * in     (DO ON BOTH SIDES)
 > A useful debug to see routes entered and removed from the BGP table is:         # debug ip bgp updates

- The prefix is in the BGP table, but not in the RIB                               # sh ip bgp | i {prefix}
 > Is the BGP next-hop reachable?                                                  # sh ip route {bgp-next-hop}

 > Is the prefix selected as the best route (indicated with '>')?                  # sh ip bgp | i {prefix}
      >> If not, verify the BGP attributes are correct.                            # sh ip bgp {prefix}

 > If a prefix is selected as best, but not entered into RIB?
      >> Could be caused by a synchronization issue!                               # sh run | i ^no synch

 > If the prefix is listed in the BGP with the options:
      >> 'r' means a lower admin distance route is used and entered in the RIB.    # sh ip bgp | i ^r.*{prefix}
      >> 's' means specific routes suppressed by aggregation are not advertised.   # sh ip bgp | i ^s.*{prefix}
      >> 'S' stale routes marked during a graceful restart is not advertised.      # sh ip bgp | i ^S.*{prefix}
      >> 'd' means the route is dampened, due to flapping violations.              # sh ip bgp | i ^d.*{prefix}

- Are any communities attached to the prefix causing problems?                     # sh ip bgp {prefix} | i entry|Community
- Are the expected communities being received? Sending communities enabled?        # sh run | i neighbor.*send-community
```

```
----------------------------------
   BGP Errors
----------------------------------
%BGP-3-NOTIFICATION: received from neighbor 196.7.8.9 2/2 (peer in wrong AS) 2 bytes 0064
      >> Local router is expecting neighbor 196.7.8.9 to come from a different AS to the configured AS-100
      >> 2 bytes 0064: The 0064 is the received ASN in HEX, i.e. 0x0064 = 100 decimal

%BGP-4-MAXPFX: No. of unicast prefix received from ...          - Approaching the max-prefix limit
%BGP-3-MAXPFXEXCEED: No. of unicast prefix received from ...    - When the max-prefix limit for a neighbor is reached
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
    OUTPUT 101
*======================*
----------------------------------
      # sh tcp brief                                                               - Shows the TCP sessions
----------------------------------
TCB       Local Address       Foreign Address         (state)
6761E614  10.5.0.1.21345      10.5.0.2.179            ESTAB
6754B140  10.5.0.1.24122      10.5.0.2.179            FINWAIT32                  - Stale session stuck in TCP handshake

----------------------------------------
      # clear tcp tcb 6754B140                                                     - Clears the stuck TCP session
----------------------------------------
[confirm]
[ok]


---------------------------------------
      # sh ip bgp summary
---------------------------------------
BGP router identifier 131.108.255.13, local AS number 1
BGP table version is 11, main routing table version 11
6 network entries and 10 paths using 854 bytes of memory
3 BGP path attribute entries using 280 bytes of memory
BGP activity 50/44 prefixes, 73/63 paths
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
131.108.1.2     4     1     194     195       11    0    0 00:03:22        2
131.108.255.6   4     1      84      83       11    0    0 00:03:23        3
131.108.255.14  4     1     152     152       11    0    0 00:03:23        3
141.199.1.1     4  1001       0       0        0    0    0 never    Idle
```

- The BGP table version is the version of the local BGP table, which is increased every time the local table is changed.
- The main routing table version shows the last version of the BGP database which was injected into the main routing table.
- The subsequent lines of text indicate the amount of memory used to store the table, and how many network known.

- Neighbor specifies the neighbor as configured on the local router.
- The version number is obvious.
- AS number of the remote neighbor.

- MsgRcvd - Number of message updates received from that neighbor since the session was established.
- MsgSent - Number of message updates that have been sent to that neighbor since the session was established.

- TblVer is used to track the changes that need to be sent to the neighbors, indicated the last table version sent to the neighbor
 > A TblVer of a neighbor that is lower than the main table indicates the neighbor is not yet fully updated.
 > Default Update internal = 30 sec eBGP and 5 sec for iBGP.

- InQ shows how many messages have been received but not processed.
 > A igh InQ could indicate lack of CPU resources to process input packets

- OutQ shows how many message are queued for delivery
 > A High OutQ could indicate lack of bandwidth to transmit packets or high CPU utilization on the other router

- Up/Down shows the time since the session was established.

- State/PfxRcd will shows the state if not established. If the session is established one will see the amount of prefix received from this neighbor.

```
---------------------------------
      # sh ip bgp
---------------------------------
BGP table version is 29, local router ID is 10.3.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale (4384514044)
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.0/14      0.0.0.0                            32768 i
* i                 10.1.13.1                0    100      0 i
s> 10.2.0.0/16      0.0.0.0                  0         32768 i
s> 10.3.0.0/16      0.0.0.0                  0         32768 i
*>i10.1.5.0/24      10.1.0.5                 0    100      0 (65002 65003) 1 i
r>i10.1.37.0/24     10.1.13.3                0    100      0 i
*>i10.1.58.0/24     10.1.0.5                 0    100      0 (65002) 1 i
*> 204.12.1.0       10.1.146.4               0             0 3 i
```

```
Status codes:
 S stale          - Indicates that the following path for the specified autonomous system is marked as "stale" during a
                        graceful restart process.
 d damped         - The table entry is dampened.
 h history        - Indicates a route that previously flapped, it has history/'baggage'.
 * valid          - This indicates valid routes
 i internal       - Indicates a prefix was learned internally via iBGP neighbor, thus it won't be advertised to other iBGP neighbors.
 > best             - This indicates a best route, candidate route to be installed in the RIB and candidate to be advertised, also
                          verifies next-hop reachability.
 s Suppressed       - Indicates more specific routes, suppressed by aggregation, that are still available in the BGP table,
                          but not advertised.
 r RIB-Failure      - Could indicate RIB-Failure.
                    - Or this could indicate that the specific prefix is already in the routing table, but with lower AD via another
                          protocol.
                    - Routes are still advertised by BGP, but not used locally. Could also point to potential routing loop.

Network Heading
 > Shows the prefixes.
 > No /prefix, indicated a classful network.

Next-hop Heading
 > This indicated the next-hop IP to reach the prefix.
 > NH of 0.0.0.0 means the prefix is directly connected.

Metric Heading
 > Indicates a MED value.
 > Value of 0 means the prefix is directly connected or no metric was configured.
 > When redistributing IGP into BGP, the IGP metric is transferred to the MED/metric field.
 > when blank indicates route was received from neighbor which has the prefix directly connected.

Locprf Heading
 > If blank, the routers default local-pref is applied, but only shown in the command 'sh ip bgp prefix'.
 > If 100, shows routes which are received from internal neighbors.

Weight Heading
 > If no weight value is specified, the default value of 0 is applied.
 > Routes which the routes originates locally/directly connected has a default value of 32768.

Path Heading
 > Indicates the AS-path list.
 > If empty the prefix is locally originated.
 > Right-most ASN indicates the originating AS.
 > Left-most ASN indicates the AS that advertised the prefix.
 > Confederation AS-path is placed in parenthesis i.e. (65002 65003).
 > The far-right letter indicated the origin codes below.

Origin Codes:
 i - IGP, network originated by using the network command, or aggregation within BGP.
 e - EGP, used to indicate routes from obsolete EGP protocol.
 ? - Incomplete, prefixes that were redistributed into BGP from an IGP or Inject-map, origin to BGP thus unknown.
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY

```
 _______     ______    _          ______
|       |   |  ___ \  | |        /  ____)
| || || |   | |___) ) | |       (  (____
| ||_|| |   |  ____/  | |        \____  \
| |   | |   | |       | |_____   _____)  )
|_|   |_|   |_|       |_______) (______/
```

```
*-=-=-=-=-=-=-=-=-=-=-*
|       INDEX         |
*-=-=-=-=-=-=-=-=-=-=-*
```

```
- MPLS Overview
      + Terminology
      + CP (Control Plane)
      + FP (Forwarding Plane)
      + RIB (Routing Information Base)
      + LIB (Label Information Base)
      + FIB (Forwarding Information Base)
      + LFIB (Label Forwarding Information Base)
- Label Operations
      + Label Structure
      + IP TTL vs Label TTL
      + TTL Propagation
      + PHP (Penultimate-Hop-Popping)
      + FEC (Forwarding Equivalence Class)
      + LSP (Label Switch Path)
      + Local and Remote Bindings
      + MPLS Label Propagation
      + MPLS Packet Forwarding
      + Aggregation
      + MTU/MRU
      + TDP (Tag Distribution Protocol)
      + LDP (Label Distribution Protocol)
            o LDP-ID (LDP-Identifier)
            o Targeted LDP Session
            o MD5 Authentication
            o Conditional Label Advertising
            o Inbound Label Filtering
- MPLS VPNs
      + MPLS VPN Terminology
      + VRF (Virtual Routing and Forwarding)
      + RD (Route Distinguisher)
      + RT (Route-Target)
      + MPLS VPN Protocols
      + MPLS VPN Label Operation
      + MPLS VPN Route Propagation
      + MPLS VPN Packet Forwarding
      + Route-Target Filter
      + VRF Import Filtering
      + Selective VRF Export
      + Hub-Spoke Scenario
      + VRF Route-Limiting
- PE to PE: MP-iBGP
- PE to CE: Connected & Static Routes
- PE to CE: RIPv2
      + Metric Transparent
- PE to CE: EIGRP
      + Using the Same AS-Number
      + Using Different AS-Numbers
      + SOO (Site Of Origin)
- PE to CE: OSPF
      + Redistributed Route Types
      + OSPF Domain-ID
      + OSPF Down-Bit
      + OSPF Domain-Tag
      + Sham-Link
- PE to CE: eBGP
      + AS-Override
      + Allowas-in
      + SOO (Site-of-Origin)
- VRF-Lite (Multi-VRF CE)
      + OSPF Capability VRF-Lite
- Troubleshooting MPLS and LDP
- Troubleshooting MPLS VPNs
- OUTPUT-101
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================*
  MPLS Overview
*=====================*
```

MPLS Overview

```
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
         > Part1:Basic MPLS
          > Multiprotocol Label Switching Overview
```

- Conventional IP routing forwards packets based on the destination IP address.
- MPLS is a highly scalable, protocol agnostic, data-carrying mechanism where data packets are assigned labels.
- Packet-forwarding decisions are made solely on the contents of the top label, without the need to examine the packet itself.
- CEF must be enabled on all MPLS routers and all MPLS interfaces.

- MPLS terminology

> LSR	- Label Switch Router, is a router that forwards packets based on labels.
> Edge-LSR	- LSR located on the edge of a MPLS network, processes both labeled and unlabeled packets.
> Ingress E-LSR	- Router that receives an unlabeled packet and inserts one or more labels before the IP header.
> Egress E-LSR	- Router that receives a labeled packet, removes all the labels and forwards it unlabeled.
> CE Router	- Customer Edge Router, a non-MPLS client/site router connected to the MPLS domain.
> Label	- A 4-byte identifier, used by MPLS to make forwarding decisions.
> Label Binding	- The mapping of a label to an FEC.
> FEC	- A group of packets forwarded in the same manner, over the same path or with the same forwarding treatment.

```
 > LSP                  - Label Switch Path, a series of LSRs that forward labeled packets to their destinations based on the FEC.
 > PHP                  - Penultimate-Hop-Popping is the act of popping/removing a label one hop before the Egress LSR/PE router.

- MPLS Components and Their Functions
 > CP (Control Plane)
      >> Describes a part of a router's architecture that is responsible for collecting and propagating the information that is
            used to forward traffic.
      >> Uses the configured routing protocols to build a routing table, often called the RIB.
      >> Uses a label exchange protocol to maintain all labels in a table called the LIB.
      >> Provides specific information from the RIB and LIB tables to the Forwarding Plane.
      >> That information is used to build the FIB and the LFIB tables.

 > FP (Forwarding Plane)
      >> Describes a part of the router's architecture that is used to decide how a packet will be forwarded once received on
            an inbound interface.
      >> Consists of two tables. The FIB and LFIB tables are responsible for forwarding incoming packets based on
             either the IP address (unlabeled) or the top label of the packet.
      >> The label functions (impose/push/insert, swap or dispose/pop/remove) happen in the FP.

 > RIB (Routing Information Base)
      >> Another name for the traditional IP routing table.
      >> Seen with "sh ip route".
      >> Table structure is: PROTOCOL, PREFIX, NEXT-HOP.

 > LIB (Label Information Base)
      >> A label exchange protocol binds locally significant labels to routes in the RIB.
      >> A label exchange protocol also exchange these label bindings among neighboring LSRs.
      >> A label exchange protocol stores local and received label bindings in the LIB table.
      >> The label exchange protocols are LDP, TDP, MP-BGP and RSVP.
      >> LDP and TDP are very similar protocols. LDP provides some additional features and is more widely used today.
      >> LDP/TDP labels are ONLY assigned to non-BGP routes in the RIB table.
      >> MP-BGP is used to distribute the label bindings for BGP routes in the routing table.
      >> RSVP is used to distribute the label bindings for TE (Traffic Engineering).
      >> The LIB table is seen with "sh mpls ldp binding" or "sh mpls ip bindings".
      >> The LIB table structure is: PREFIX, LSR/LOCAL, LABEL.

 > FIB (Forwarding Information Base)
      >> A CEF built table sourced from the information in the RIB table and then used to forward incoming IP packets.
      >> An arriving IP packet is forwarded unlabeled (as an IP packet) if no label for the destination route exists.
      >> An arriving IP packet is forwarded labeled if a next-hop label is available for the destination route.
      >> The FIB table is seen with "sh ip cef detail".
      >> The FIB table structure is: PREFIX, NEXT-HOP, LABEL.

 > LFIB (Label Forwarding Information Base)
      >> A CEF built label table soured from the information in the LIB.
      >> The LFIB table ONLY stores the labels used to forward packets, unlike the LIB table that stores ALL label bindings.
      >> The 'incoming label' is a 'local label' that is advertised to adjacent LSRs.
      >> The 'outgoing label' is the received 'local label' from the next-hop LSR to a destination.
      >> The LFIB table is seen with "sh mpls forwarding table".
      >> The LFIB table structure is: INLABEL, OUTLABEL, NEXT-HOP.
```

```
----------
 COMMANDS
----------
# sh ip route                                   - Shows the RIB table
# sh ip cef [detail]                            - Shows the FIB table
# sh mpls ldp bindings                          - Shows the LIB table
# sh mpls ip bindings                           - Shows the LIB table
# sh mpls forwarding-table                      - Shows the LFIB table

                                                >>> Configuring basic MPLS <<<
#ip cef                                   Step1 - Enables CEF which is a pre-requisite (default=enabled)
#mpls label protocol [ldp|tdp|both]       Step2 - Selects a label distribution protocol to be used
                                                - From IOS 12.4(3) LDP is default

#interface (int)
 #mpls ip                                 Step3 - Enables label switching (starts TDP/LDP) on an interface
                                                - 'tag-switching ip' is the older syntax
```

-=-

```
*======================*
   Label Operations
*======================*
```

- Layer2 frames have an NLPID (Network Layer Protocol Identifier) field to indicate what the layer3 payload is.
- An MPLS label, often called a SHIM header, is inserted between the layer2 and layer3 headers.
- As a result, a router must change the layer2 NLPID or ethertype to indicate that the packet is an MPLS-labeled packet and not a standard layer3 packet.

- Labels are 4-byte identifiers used for forwarding decisions.
- Each MPLS label is 4-bytes/32-bits and has the structure:

```
      0             19    22  23       31
      +------------------+-----+---+---------+
      |                  |     |   |         |
      |     L A B E L    | EXP | S |   TTL   |
      |                  |     |   |         |
      +------------------+-----+---+---------+
```

> 20-bit label	- Actual label value (labels 0 to 15 are reserved).
> 3-bit experimental field	- Used to define a class of service. RFC 5462 renamed this to TC (Traffic Class).
> 1-bit bottom-of-stack	- Indicates the last label in the label-stack (1=true, 0=false).
> 8-bit TTL	- MPLS label TTL, used to prevent loops.

```
- Label TTL
 > By default, on MPLS label imposition, the IP TTL is decremented and propagated/copied to the label TTL for loop prevention.
 > At every hop in the MPLS network, only the top label's TTL is decremented.
 > If the top label is swapped, the label TTL from the arriving top label is decremented before being copied to the swapped label.
 > If the top label in a label stack is popped, the label TTL from the top label is propagated/copied to the next label.
 > If an additional label is imposed, the TTL from the arriving top label is decremented and copied to the swapped label's TTL
     as well as the newly-added top label's TTL.
 > At MPLS label disposition the top label's TTL is copied to the IP TTL, but ONLY if the label TTL < IP TTL.
 > An LSR only looks at or only changes the top label in the label stack of an arriving packet.
 > The underlying labels and the IP packet's TTL are left unchanged.

- TTL Propagation
 > TTL propagation is enabled by default.
 > Disabling TTL propagation can be used to hide core LSRs in an MPLS backbone.
 > If disabled, the ingress label TTL is set to 255 on MPLS label imposition.
 > If fully disabled, the core LSRs will not show up in a traceroute done from a edge-LSR or a client CE router.
 > If disabled only for forwarding traffic:
      >> A traceroute done from the edge-LSR will show the core LSRs.
      >> A traceroute done from the client CE router will not show the core LSRs.

- PHP (Penultimate-Hop-Popping)
 > A egress LSR advertises a label value of 3 (IMP-NULL) to an upstream (penultimate) LSR, instructing that LSR to pop the
      top label, before forwarding the packet on to the egress LSR.
 > PHP removes the requirement for a double lookup to be performed on an egress LSR.
 > The LIB table will display a value of imp-null.

- Labels define the destination and services for a packet, and identify the FEC (Forwarding Equivalence Class).
- Labels have local significance, because each LSR independently maps a label to an FEC in a label binding.
- Label bindings are usually only exchanged between adjacent LSRs.

- FEC (Forwarding Equivalence Class)
 > Is a flow of packets that are forwarded along the same path or that share the same forwarding treatment.
 > All packets belonging to the same FEC have the same label.
 > However not all packets that have the same label belong to the same FEC, because the EXP values might differ.
 > The ingress LSR classifies and assigns packets to a specific FEC using a label.
 > By default no further packet classification is done in the MPLS network.
 > A FEC can correspond to any of the following:
      >> An MPLS unicast IP traffic FEC corresponds to a destination network stored in the RIB.
      >> An MPLS multicast IP routing FEC is equal to a destination multicast address.
      >> An MPLS VPN FEC is equal to the VPN routing table on the BGP next-hop.
      >> An MPLS QOS FEC is equal to a combination of a destination network and a COS (Class of Service) value.
 > The remainder of this section will focus on the MPLS unicast IP FEC where a label = prefix.
 > The MPLS VPN Section will cover MPLS VPN FEC where the BGP next-hop is the FEC.

- LSRs can perform the following label operations:
 > Insert (impose) a label.
 > Swap a label.
 > Remove (pop) a label.

- Label Stack
 > When more than one label is inserted between the layer2 and layer3 headers.
 > The first label in the stack is called the top (outer) label, and the last label is called the bottom (inner) label.
 > Label forwarding decisions are made based ONLY on the top label in a stack.
 > With basic MPLS the label stack only consists of one label. MPLS VPNs, TEs and ATOMs use multiple labels.
 > Copy descretly owned by Kane  Bagwell
```

- LSP (Label Switch Path)
 > Series of LSRs that forward labeled packets based on the FEC.
 > LSPs are unidirectional; return traffic will follow a different LSP.
 > LDP only advertises labels for the individual segments in the LSP.

- Local Bindings
 > A LSR assigns one locally significant label per route in the RIB table.
 > The local binding is this one route and its associated label. The local bindings are stored in the LIB table.
 > This label is typically referred to as the local/IN label depending on which table is referenced.
 > Every LSR will advertise its local bindings to its adjacent LSRs using TDP/LDP.
 > The local bindings that a LSR advertises, tell other LSRs what label values it expects traffic to arrive with.
 > A LSR will only accepts labeled packets if the top labels in the label stack match the local labels it previously advertised.
 > If an LSR receives a labeled packet with a top label, that is not one of its own local labels, it will drop the packet.

- Remote Bindings
 > A received binding advertised by an adjacent TDP/LDP LSR.
 > A TDP/LDP LSR may receive multiple label bindings for each route, usually one per TDP/LDP peer.
 > All these received bindings (remote bindings) are stored in the LIB.
 > Only one LSR can be the downstream LSR for a particular route, unless load balancing is configured.
 > The downstream LSR is the next-hop for a particular route in the RIB.
 > Only the remote binding received from the next-hop LSR is used to populate the OUT label in the LFIB.

- Downstream is always towards the destination as indicated by the RIB.
- Upstream is always towards the source as indicated by the RIB.

- MPLS Label Propagation
 > The configured IGP converges as normal and advertises all the routes to all the neighbors.
 > The routes are populated in the RIB and FIB tables.
 > Each LSR assigns a label (local/IN label) to each non-BGP route in its RIB table and stores it in the LIB and LFIB tables.
 > Each allocated local binding is independently advertised to all neighbor LSRs, regardless of whether the neighbors are upstream or downstream for a particular destination.
 > Upon receiving the label advertisements, each LSR stores all the remote bindings and details of the LSRs that advertised each one in their LIB tables.
 > After label convergence, each LSR processes its RIB table to find the next-hop/downstream LSR for each destination.
 > The label (next-hop/OUT label) received from the next-hop/downstream LSR for each destination is taken from the LIB and placed into the FIB and the LFIB.
 > Once the LFIB is fully populated LDP is considered converged.
 > For each route, the LSR always has a single local binding (IN label) and one remote binding per LDP peer (OUT labels).
 > SUMMARY: For label forwarding to take place, locally-assigned labels that were previously advertised to upstream neighbors are mapped to next-hop labels, previously received from downstream neighbors.

- MPLS Packet Forwarding
 > If a LSR receives an IP packet, the FIB is used to make the forwarding decision:
 >> If the next-hop IP has an OUT label associated, the packet is labeled and forwarded to the next-hop LSR.
 >> If the next-hop IP does not have an OUT label associated, the packet is forwarded unlabeled to the next hop.
 >> If there is no next-hop available for the destination, the packet is dropped.

```
 > If a LSR receives a labeled packet, the LFIB is used to make the forwarding decision.
      >> If the top label matches an IN label, the corresponding OUT label determines if:
            >>> The top label is swapped before the packet is forwarded.
            >>> The top label is swapped and another label imposed before the packet is forwarded.
            >>> The top label is popped and the packet is forwarded labeled if any labels remain in the label stack
                or is forwarded unlabeled if the last label in the stack was removed.
      >> If the top label does not match any of the IN labels, the packet is dropped.
            >>> This is the case regardless, even if the FIB has the IP destination listed.

- Aggregation/Summarization
 > An aggregating LSR will advertise the labels for the routes it is summarizing.
 > The outgoing labels in the LFIB will show 'aggregate'.
 > If an LSR receives a labeled packet for routes it is aggregating, it will remove the label stack.
 > The LSR will then do an IP lookup for the more specific route and find the new outgoing label.
 > The IP packet will then be labeled before being forwarded on.
 > Aggregation in MPLS networks breaks end-to-end LSPs and often causes traffic to be dropped.
 > For this reason LSR loopback addresses should never be summarized.

- MTU (Maximum Transmission Unit)
 > Indicates the maximum size of an IP packet that can be transmitted on a link without fragmenting the packet.
 > IP MTU is the maximum size a layer3 IP packet can be without requiring fragmentation.
 > MPLS MTU is the maximum size a labeled packet (IP packet + label/s) can be without requiring fragmentation.
 > By default, the MPLS MTU value is derived from the IP MTU value. This is usually a problem on ethernet links.
 > How does the fragmentation in MPLS work?
      >> If a labeled packet is received by a LSR that notices the outgoing MTU is not big enough for the packet,
            the LSR strips off the label stack, fragments the IP packet, puts the label stack onto all the fragments
            and forwards the fragments.
      >> The only exception is when the DF (Don't Fragment) bit is set in the IP header, which will generate
            an ICMP type 3,code 4 message.

 > The typical MTU size on an ethernet interface is 1500-bytes.
 > Since each label is 4-bytes, a packet's size increases with 4-bytes for every label added.
 > If one label is added to an already maximum sized IP packet of 1500-bytes the packet will be fragmented.
 > To prevent this typical fragmentation problem, either increase the MPLS MTU if possible or decrease the IP MTU.
 > Another alternative is to enable PMTU to auto discover the maximum allowed unfragmented IP MTU to the endpoint.
 > The MPLS MTU value is configured with "mpls mtu" under the interface.
 > The configured interface values can be seen with "sh int | i MTU" and "sh mpls int detail | i MTU".

- MPLS MRU (Maximum Receive Unit)
 > The maximum size a received labeled packet may be without the need to be fragmented when forwarded out of
      the egress interface.
 > This value is derived from the MPLS MTU on the egress interfaces.
 > Scenario:
      >> Assume the MPLS MTU is set to 1504-bytes on all interfaces. This allows a single label on packet leaving the LSR .
      >> If a received packet's label operation is 'pop', the MRU for that route on that router would be 1508-bytes.
      >> This is because when the packet is sent out, it can be no bigger than 1504-bytes.
      >> But since the operation is POP, the packet may be received with 2 labels (1508-bytes)
      >> One label will be popped and the sent packet size of 1504-bytes would be allowed.
 > The MRU value takes into account the amount of labels pushed, swapped, or popped on the local router.
```

> This value is calculated per FEC and not just per interface.
> The MRU value can be seen in the LFIB with "sh mpls forwarding-table detail | i MRU".

- There are different protocols that distribute labels:

> TDP is used to distribute the bindings for non-BGP routes in the routing table.
> LDP is used to distribute the bindings for non-BGP routes in the routing table.
> RSVP distributes MPLS TE labels, which is beyond the scope of R&S.
> MP-BGP is used to distribute the bindings for BGP routes in the routing table.

- The terms TDP and LDP are often used synonymously, but the differences are as follows.

- TDP (Tag Distribution Protocol)

> A Cisco proprietary protocol.
> Refers to labels as tags.
> Uses local broadcasts instead of multicast.
> Uses UDP and TCP port 711.
> Does not support MD5 authentication.

- LDP (Label Distribution Protocol)

> DOC-CD LOCATION
> 12.4T Configuration Guides
> Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
> Part2:MPLS Label Distribution Protocol
> MPLS Label Distribution Protocol (LDP)

> An IETF standards protocol.
> Uses an all-routers multicast address (224.0.0.2) for directly-connected neighbor discovery.
> Uses UDP and TCP port 646 for neighbor discovery and session establishment.
> An LDP session is established from the LSR with the higher IP address.
> Provides optional MD5 authentication.
> The LDP hello interval is every 5 sec and the hold interval is 15 sec.
> If two LDP neighbors have different values configured, the smaller of the two is used.
> LDP sessions should be established between routable loopback addresses of an adjacent pair of LSRs.

- LDP-ID (LDP-Identifier)

> A 6-byte field consisting of 4-bytes identifying the LSR and 2-bytes identifying the label space.
> The first 4-bytes is similar to any router-id. This is taken from the operational interface with the highest IP address, however, if a loopback is defined, the loopback with the highest IP address is used.
> If the last two bytes are zero, the specific details are beyond the scope of the CCIE R&S.
> The LDP-id can also be manually configured with the "mpls ldp router-id force" command.
> Important to remember the LDP-id must be routable, else LDP sessions will not be established.

- Targeted LDP Session

> Is required with non-adjacent LDP neighbors.
> The UDP hellos are sent to a specified unicast IP address instead of the multicast IP address.
> When the neighbor is discovered, the mechanism to establish a session is the same.
> Configured with "mpls ldp neighbor {ip} targeted".

```
- LDP Authentication
 > To protect LDP sessions MD5 authentication can be configured between neighbors.
 > MD5 authentication adds a signature (the MD5 digest) to the TCP segments. The MD5 digest is calculated based on the password.
 > Only the MD5 digest is transmitted, the passwords are never transmitted.
 > MD5 authentication is required on both sides of a link between two neighbors.
 > Configured with "mpls ldp neighbor {ip} password".
 > If one LSR has MD5 configured for LDP and the other not, the following message will be logged:
      %TCP-6-BADAUTH: No MD5 digest from 10.5.1.4(11092) to 10.5.1.3(646)
 > If both LDP neighbors have MD5 configured but the passwords don't match, the following message will be logged:
      %TCP-6-BADAUTH: Invalid MD5 digest from 10.5.1.4(11093) to 10.5.1.3(646)
```

```
- Conditional Label Advertising
 > Enables the selective advertisement of only the necessary labels to certain LDP neighbors.
 > The 'FOR' keyword specifies which routes should have their labels advertised.
 > The optional 'TO' keyword specifies which LDP peers should receive the label advertisements. This must match the LDP router-id.
 > More than one "mpls ldp advertise-labels" statement can be used on the same LSR.
```

```
CONFIG-SET: Conditional Label Advertising for the Loopback's IPs
+-----------------------------------------------------------------
|     no mpls ldp advertise-labels                           - Disables the default behavior to advertise all labels
|     access-list 10 permit 10.5.1.0 0.0.0.255               - Matches all loopback addresses
|     access-list 11 permit any                              - Matches any neighbor
|     mpls ldp advertise-labels for 10 to 11                 - Labels matching ACL-10 are send to neighbors matching ACL-11
|
```

```
- LDP Inbound Label Binding Filtering
 > DOC-CD LOCATION
       > 12.4T Configuration Guides
        > MPLS
         > Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
          > Part2:MPLS Label Distribution Protocol
           > MPLS LDP Inbound Label Binding Filtering

 > Allows filtering incoming label bindings from a LDP neighbor.
 > If you are bored this could be used to limit the number of label bindings stored in the LIB.
```

```
CONFIG-SET: Filtering Inbound Label Bindings
+-------------------------------------------------
|     access-list 12 permit host 10.5.1.250                  - Specifies each label binding that should be accepted
|     access-list 12 permit host 10.5.1.251
|     !
|     mpls ldp neighbor 10.5.1.250 labels accept 12          - Filters label binding matching ACL-12 from neighbor 10.5.1.250
|
------------
 COMMANDS
------------
# sh mpls interface {int} [detail]              - Shows the MPLS enabled interfaces, their status, and MTU settings
# sh mpls label range                           - Shows the range of local labels available

# sh int {int} | i MTU                          - Shows the IP MTU per interface
# sh mpls interfaces [int] detail | i MTU        - Shows the MPLS MTU per interface
# sh mpls forwarding {route} detail | i MRU     - Shows the MRU for the specified route
# sh mpls forwarding labels {label} exact-path ipv4 {src-ip} {dst-ip}
                                                - Shows the exact exit link a labeled IPv4 packet will take based
                                                            on the address pair

# sh ip cef [vrf] exact-route {src-ip} {dst-ip} - Shows the exact exit link a packet will take based on
                                                            the address pair

# sh ip cef table [vrf]                         - Shows information about the CEF tables and amount of routes
# sh ip cef switching statistics                - Useful if CEF is dropping IP packets
# sh adjacency                                  - Shows the CEF adjacency table, including the NLPID

# clear cef table                               - Refreshes the CEF cache
# clear adjacency                               - Clears the layer2 rewrite information

# debug ip cef drops [acl]                      - Shows if CEF is dropping IP packets on a ingress LSR

# debug mpls lfib                               - Debugs LFIB events: label creations, removals, rewrites
# debug mpls packet [mpls-acl] [interface]      - Debugs labeled packets switched by router
# debug mpls ldp                                - Debugs LDP adjacencies, session establishments, label binding exchanges

#mpls ldp router-id {interface} [force]         - Configures the MPLS-ID, (interface must be in an up state to be used)
                                                - [force]: Forcibly changes the router-id before a reload

#mpls ldp neighbor [vrf {name}] {ip} targeted   - Establishes a targeted LDP session with nonadjacent neighbor.
#mpls label range [low high]                    - Changes the default label range (16-100000)
#no mpls ip propagate-ttl [forwarded|local]     - Disables TTL propagation, useful to hide core LSRs (default = enabled)
                                                - Forwarded: Trace doesn't work for transit traffic labeled by the router
                                                - Local: Trace doesn't work from the router, but transit traffic does

#system jumbomtu {bytes}                        - Enables jumbo MTUs on 3560 switches
#interface fa0/0
 #ip mtu {bytes}                                - Max size a layer3 IP packet can be without requiring fragmentation.
                                                - The interface MTU is automatically increased on WAN interfaces; IP MTU is
                                                      automatically decreased on LAN interfaces
                                                - Min MTU is 64 bytes, Max MTU depends on the interface type
 #mpls mtu {bytes}                              - Max size a labeled packet can be without requiring fragmentation.
                                                - (default = IP MTU)


#mpls label protocol [ldp|tdp|both]             - Selects a label distribution protocol to be used
#mpls ldp neighbor [vrf] {ip} password {pwd}    - Configures a MD5 password authentication for LDP
```

```
#no mpls ldp advertise-labels                    - Disables default behavior to advertise all labels to all neighbors
#mpls ldp adv-labels [for {prefix-acl}] [to {peer-acl}]
                                                 - Configures conditional label advertising
                                                 - [for]: Specifies the destinations for which labels are generated
                                                 - [to]: Specifies a recipient list of neighbors

#mpls ldp neighbor [vrf] {ip} labels accept {acl}
                                                 - Configures filtering inbound LDP label bindings


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================*
  MPLS VPNs
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
         > Part7:MPLS Layer 3 VPNs
          > Configuring MPLS Layer 3 VPNs

- Defined in RFC-4364, previously RFC-2547.
- A MPLS VPN combines BGP with multi-protocol extensions, MPLS traffic isolation and router support for VRFs to create an
      IP-based VPN.

- MPLS VPN Terminology
 > Label                  - A 4-byte identifier, used by MPLS to make forwarding decisions.
 > CE Router              - Customer Edge Router, a non-MPLS client/site router connected to the MPLS domain.
 > P Router               - Provider Router, a LSR in MPLS VPN terminology.
 > PE Router              - Provider Edge Router, an edge-LSR in MPLS VPN terminology.
 > LSP                    - Label Switch Path, a series of LSRs that forward labeled packets to their destinations. (unidirectional)
 > Ingress PE router      - Is the edge-LSR an IP packet arrives at from a CE router before being labeled and forwarded to
                                the egress PE router.
 > Egress PE Router       - Is the edge-LSR where the destination route is connected. Receives labeled packets, forwards IP packets.


- VRF (Virtual Routing and Forwarding)
 > Is a technology that allows multiple instances of tables to co-exist on the same router.
 > Each instance operates independently and provides isolation between different clients running the same address space.
 > A VRF consists of a separate RIB, FIB, and LFIB table per instance.
 > A VRF is locally significant to a router.
 > Traffic that enters on a VRF enabled interface will belong to that VRF instance.
 > Each interface can only be assigned to one VRF, but a VRF can have many interfaces assigned.


- RD (Route Distinguisher)
 > The VPN routes are propagated across a MPLS VPN network by MP-iBGP. But MP-iBGP requires that the transported routes be unique.
 > An RD is a 64-bit (8-byte) value prepended to a client's non-unique 32-bit IPv4 address to produce a unique 96-bit VPNv4 address.
 > An RD uniquely identifies a route (IP route), it does NOT identify a VPN.
 > An RD is locally significant to a router but has global relevance.
```

```
- RT (Route-Target)
 > Is a 64-bit (8-byte) extended BGP community that is attached to a VPNv4 BGP route to indicate its VPN membership.
 > Any number of RTs can be attached to a single route up to the BGP Update packet size of 4096.
 > Export RTs
      >> Are attached to a route when it is converted into a VPNv4 route.
      >> Identify the VPN membership by associating routes to a VRF.
 > Import RTs
      >> Are used to select VPNv4 routes for insertion into matching VRF tables.
      >> On the receiving PE router, a route is imported into a VRF only if at least one RT attached to the route matches at least
            one import RT configured in that VRF (route-map conditions must be met, if configured).
 > An import or export map allows route control on a per-route basis.
 > RTs allow for more complex VPN designs like Hub-and-Spoke, Central Services, Extranet, Management VPNs, etc.


- RT/RD can be used in one of the following formats:
 > ASN:nn                 - Autonomous System Number; where 'nn' can be any number.
 > IP-ADD:nn              - 4-Octet Dotted Decimal format; where 'nn' can be any number.


- Loopback Interfaces
 > With MPLS VPNs it is almost a requirement to use loopback interfaces on all P and PE routers.
 > These loopbacks must be included in the core IGP (e.g. OSPF).
 > The MP-BGP sessions should be set up using these loopback addresses to avoid premature label popping in LSPs.
 > These loopback interfaces will be used and referred to as the BGP next-hop address to carry MPLS VPN traffic.
 > A BGP next-hop address must be an IGP route.


- Protocols required for MPLS VPNs:
        .---.           .---.            .--.            .---.            .---.
      | CE1 |----------| PE1 |----------| P1 |----------| PE2 |----------| CE2 |
        '---'           '---'            '--'            '---'            '---'
         |-----IGP1-----|  |----IGP2------||-----IGP2-----|  |----IGP1-----|
                           |-----LDP------||-----LDP------|
                           |------------MP-iBGP-----------|

 > IGP1             - This is a per-VRF IGP to advertise client routes in the VRF routing table.
                    - IGP1 could be any of the following: statics routes, RIP, EIGRP, OSPF, eBGP.
 > IGP2             - This is the core MPLS IGP. This is generally OSPF or IS-IS.
 > LDP              - Either TDP or LDP could be used as the label exchange protocol between all MPLS-enabled routers.
 > MP-BGP           - For MPLS VPNs MP-BGP sessions are only required between PE routers (refer to MP-iBGP Section below).


- With MPLS VPNs, two labels are used in the label stack:
 > The outer/top label is used for switching the packet through the MPLS network (often called the LDP label).
 > The top label points to the egress router and is propagated by LDP (the adjacent LSR's label for the next-hop's IPv4 route).
 > The inner/bottom label is used to separate packets at egress points (often called the VPN label).
 > The second label identifies the outgoing interface on the egress router and is propagated via MP-BGP.

- MPLS VPN Label Operation
 > The configured IGP converges as normal, advertising the BGP next-hop IPs.
 > TDP/LDP converges as described in the previous Section, advertising the TDP/LDP labels for the BGP next-hops.
 > Every egress PE router assigns a VPN label to every local VRF route.
 > MP-iBGP on the PE routers converges by advertising all local VRF routes along with the VPN labels to ALL other PE routers
      in MP-iBGP updates.
 > Once converged, all PE routers should have an OUT VPN label assigned to each non-local VRF route along with a LDP label for
      every BGP next-hop.
 > These two labels per route (VPN label, LDP label) are the two labels MPLS VPN uses in a label stack.
```

```
- MPLS VPN Route Propagation
 > This scenario depicts how a route is advertised from CE2 to CE1 across a MPLS VPN network.

       .---.            .---.            .--.            .---.            .---.
      | CE1 |----------| PE1 |----------| P1 |----------| PE2 |----------| CE2 |---| 10.5.1.0/24
       '---'            '---'            '--'            '---'            '---'
                      <--7)(6)(5)                      <--4)(3)(2)      <--1)

 1- CE2 advertises an IPv4 route (10.5.1.0/24) to PE2, using the configured IGP or eBGP.
 2- The received IPv4 route advertisement is inserted into the VRF routing table associated with the PE2 interface it arrived on.
 3- When the IPv4 route is redistributed into MP-BGP, PE2 prepends the 64-bit RD from the VRF to the non-unique 32-bit IPv4 route,
      which results in a globally unique 96-bit VPNv4 packet. At the same time the export RTs from the VRF are attached.
 4- The VPNv4 route (with VPN label and RTs) is advertised via MP-iBGP sessions to all other PE routers (PE1 in this scenario).
 5- PE1 receives the MP-BGP update but only processes the VPNv4 route if it is configured locally. If configured locally, implying
      the RTs attached to the route match an import RT in a local VPN, the VPNv4 route is imported into the appropriate
      VPNv4 BGP table.
 6- PE1 then strips the RD off the VPNv4 route, and installs the original IPv4 route (10.5.1.0/24) into the VRF table.
 7- PE1 advertises the IPv4 route (10.5.1.0/24) via the configured IGP or eBGP to CE1
```

```
- MPLS VPN Packet Forwarding
 > This scenario depicts how a packet is routed from CE1 to CE2 across a MPLS VPN network.

       .---.            .---.            .--.            .---.            .---.
      | CE1 |----------| PE1 |----------| P1 |----------| PE2 |----------| CE2 |---| 10.5.1.0/24
       '---'            '---'            '--'            '---'            '---'
             (1-->      (2)(3)(4-->        (5-->            (6) (7-->

 1- CE1 sends an IPv4 packet destined to 10.5.1.0/24, towards PE1.
 2- The ingress PE router (PE1) receives the packet and looks up the next-hop for the destination in the VRF routing table
      associated with the ingress interface the packet arrived on. The egress PE router (PE2), which previously advertised this
      route (and a VPN label), will be used as the next-hop.
 3- Since a label was received from the next-hop, the packet will be labeled:
      - The bottom label is the VPN label, which will be used to indicate the correct CE next-hop on PE2.
      - The top label will be the LDP used to get to PE2 loopback. On PE1 this would be a LDP label received from P1.
 4- The labeled packet is forwarded to P1.
 5- P1 receives a labeled packet, checks the LFIB table and pops (PHP) the LDP label before forwarding the labeled packet to PE2.
 6- PE2 receives a labeled packet, with the top label matching a VPN route pointing to the IP next-hop, CE2.
 7- The packet is forwarded unlabeled as a IPv4 packet towards CE2.
 8- Return traffic will follow the same process but in reverse (remember a LSP is unidirectional).
```

```
!!NOTE!!- Always make sure that the VPN label is only exposed on egress PE routers where the VRF is configured, otherwise PHP will
      occur prematurely and traffic will be dropped.


CONFIG-SET: Simple Full-Mesh VPN between the two sites connected to two PE routers
+-----------------------------------------------------------------------------------------
|PE1#
|     ip vrf BOB
|     rd 123:1
|     route-target export 123:1                              - Exports all VRF-RIB routes with a RT of 123:1
|     route-target import 123:1                              - Imports MP-BGP routes if the RT of 123:1 matches
|     !
|     interface serial2/4                                    - The interface connected to CE1
|     ip vrf forwarding BOB                                  - Assigns the interface to VRF-BOB
|
|PE2#
|     ip vrf BOB
|     rd 123:1
|     route-target export 123:1                              - Exports all VRF-RIB routes with a RT of 123:1
|     route-target import 123:1                              - Imports MP-BGP routes if the RT of 123:1 matches
|     !
|     interface serial3/2                                    - The interface connected to CE2
|     ip vrf forwarding BOB                                  - Assigns the interface to VRF-BOB
|
```

- Route-Target Filter
 > LSRs by default only process MP-BGP advertisements for VRFs that are locally configured (VRF import statement).
 > The other advertisements are ignored and not entered into any table.
 > This default behavior can be disabled with "no bgp default route-target filter".
 > Since RRs (Route Reflectors) don't have the VRFs configured locally, by default RRs will not accept any VPNv4 routes.
 > The route-target filter must be disabled on RRs.

- Also keep in mind that RRs only reflect the best routes.

- VRF Import Filtering
 > By using default configuration all routes matching an import RT will be imported.
 > A VRF import route-map allows more granularity by only importing selected routes.
 > A route is only imported into a VRF if at least one RT attached to the route matches one RT configured in the VRF and the route is accepted (permitted) by the import route-map.
 > The route-map can match routes using the following criteria:
      >> Access-lists
      >> Prefix-lists
      >> RTs.
 > The route-map can only be configured in addition to a RT import statement "route-target import {rt}".
 > If a VRF import route-map is configured, routes must be explicitly allowed for import. If a route did not match any route-map instance it will be imported and filtered.

```
CONFIG-SET: MPLS-VPN - VRF Import Filtering Example
+--------------------------------------------------------
|     access-list 55 permit 10.5.1.0 0.0.0.255              - Matches a specific route
|     !
|     ip extcommunity-list 10 permit rt 123:2               - Creates a community-list matching RT 123:2
|     !
|     route-map IMPORT permit 10
|      match extcommunity 10                                - Routes with a RT of 123:2 will be imported
|     !
|     route-map IMPORT deny 20
|      match ip address 55                                  - The route 10.5.1.0/24 will be imported
|     !
|     route-map IMPORT permit 30                            - Allows all other routes matching 123:789 to be imported
|     !
|     ip vrf CLIENT-A
|      rd 123:789
|      import map IMPORT                                    - Applies the import-map, importing ALL routes with a RT 123:2
|                                                                 and 10.5.1.0/24 if its RT is 123:789
|      route-target import 123:789                          - Imports all MPBGP routes with a RT of 123:789
|      route-target export 123:789                          -     Exports all VRF CLIENT-A RIB routes with a RT of 123:789
```

```
- Selective VRF Export
 > By default all routes in the VRF RIB will be exported with the default export RTs.
 > A VRF export route-map can be used to achieve any of the following:
      >> Only export selective routes to the MP-BGP table for advertisement.
      >> Attach extra RTs in addition to the default RTs (often used in extra-net designs).
 > The implicit 'no-match' at the end of a route-map DOES NOT prevent the route from being exported. If a route did not match
      any route-map instance it will be exported using the default route-target export.
 > An explicit deny in an export-map will prevent a route from being exported.
 > An export-map command with a "set extcommunity rt" command clears already added RTs. If the additive keyword is specified that
      RT is added in addition to the already-set RTs.
 > Selective VRF export does NOT require a RT export statement if " set extcommunity rt" is configured.
 > The following two config-sets accomplishes the same tasks:
      >> 20.1.20.0/24 is not exported.
      >> 10.5.1.0/24 is exported with two RTs.
      >> All other routes are exported with one RT.
```

```
CONFIG-SET: MPLS-VPN - Selective VRF Export Option-1
+------------------------------------------------------
|     access-list 55 permit 10.5.1.0 0.0.0.255              - Matches a route to be exported
|     access-list 66 permit 20.1.20.0 0.0.0.255             - Matches a no-export route
|     !
|     route-map EX-MAP deny 10
|      match ip address 66                                  - Explicitly prevents 20.1.20.0/24 from being exported
|      !
|     route-map EX-MAP permit 20
|      match ip address 55                                  - References ACL-55 for routes to be exported
|      set extcommunity rt 123:555 additive                 - Adds RT 123:55 additionally onto 10.5.1.0/24
|     !
|     ip vrf CLIENT-B
|      rd 123:789
|      export map EX-MAP                                    - Applies the export-map
|      route-target import 123:789                          - Imports all MP-BGP routes with a RT of 123:789
|      route-target export 123:789                          - All VRF CLIENT-B RIB routes not matched by the EX-MAP are
|                                                                exported with a RT of 123:789
|
```

```
 CONFIG-SET: MPLS-VPN - Selective VRF Export Option-2
+------------------------------------------------------
|     access-list 55 permit 10.5.1.0 0.0.0.255              - Matches a global route
|     access-list 66 permit 20.1.20.0 0.0.0.255             - Matches a no-export route
|     !
|     route-map EX-MAP deny 10
|      match ip address 66                                  - Explicitly prevents 20.1.20.0/24 from being exported
|      !
|     route-map EX-MAP permit 20
|      match ip address 55                                  - References ACL-55
|      set extcommunity rt 123:555 123:789                  - Attaches RT 123:55 and RT 123:789 to 10.5.1.0/24
|      !
|     route-map EX-MAP permit 30
|      set extcommunity rt 123:789                          - All other routes have RT 123:789 attached
|     !
|     ip vrf CLIENT-B
|      rd 123:789
|      export map EX-MAP                                    - Applies the export-map
|      route-target import 123:789                          - Imports all MP-BGP routes with a RT of 123:789
|
```

- Hub-Spoke Scenario
 > A hub-spoke design could be used when full connectivity between sites is prohibited, or if there is a need, say security, for all branch-to-branch traffic to flow via the head office site.
 > It is unlikely that this will be seen in production, but it is nice to know for the lab.

```
CONFIG-SET : MPLS-VPN Hub-Spoke Design Example with a Pitfall
+---------------------------------------------------------------
|Example: Three client sites, all communication must traverse the HUB-site.
|     Site-1 and Site-2 connects to the same PE2 router.
|     The HUB-site connects to PE1, and Site-3 connects to PE3.
|
|PE1#
|
|     ip vrf BOB                                            - Creates the locally significant VRF tables named BOB
|     description THE-HUB_SITE
|     rd 123:1
|     route-target export 123:100                           - Exports the BOB-HQ routes
|     route-target import 123:200                           - Imports the routes from all BOB's sites
|     !
|     interface serial3/2
|     ip vrf forwarding BOB                                 - Assigns Serial3/2 to VRF-BOB
|
|PE2#
|     ip vrf BOB                                            - Creates the locally significant VRF tables named BOB
|     description SITE-1
|     rd 123:2
|     route-target export 123:200                           - Exports the SITE's routes
|     route-target import 123:100                           - Imports the HQ routes from BOB-HQ
|     !
|     interface serial1/1
|     ip vrf forwarding BOB
|
|     ip vrf BOB-2                                          - HERE IS THE CATCH. A separate set of VRF tables are needed,
|     description SITE-2                                        otherwise Site-1 and Site-2 will share VRF-BOB and
|     rd 123:22                                                 thus be allowed to communicate directly
|     route-target export 123:200                           - Exports the SITE routes
|     route-target import 123:100                           - Imports the HQ routes from BOB-HQ
|     !
|     interface serial1/1
|     ip vrf forwarding BOB-2
|
|PE3#
|     ip vrf BOB                                            - Creates the locally significant VRF tables named BOB
|     description SITE-3
|     rd 123:3
|     route-target export 123:200                           - Exports the SITE routes
|     route-target import 123:100                           - Imports the HQ routes from BOB-HQ
|     !
|     interface serial5/1
|     ip vrf forwarding BOB
```

```
- VRF Route-Limiting
 > The number of routes within a VRF table can be limited explicitly.
      >> This applies to all routes on a router within the VRF, not just BGP routes.
      >> This applies to routes learned from CE routers and other PE routers.
      >> The default behavior when the limit is reached is that the router won't accept anymore VRF routes.
      >> Log messages:  %IPRT-3-ROUTELIMITWARNING: IP routing table limit warning....
                        %IPRT-3-ROUTELIMITEXCEEDED: IP routing table limit exceeded....

 > The number of routes received from a BGP neighbor could be limited.
      >> The default behavior when the limit is reached is to drop the neighbor relationship.
      >> Log messages:  %BGP-4-MAXPFX: No. of unicast prefix received from ...
                        %BGP-3-MAXPFXEXCEED: No. of unicast prefix received from ...
```

```
----------
 COMMANDS
----------
# sh ip vrf                                     - Shows the list of all VRFs configured in the router
# sh ip vrf [detail]  {vrf-name}                - Shows the VRFs configured and associated interfaces
                                                - [detail] Displays the import/export parameters per VRF
# sh ip vrf interface                           - Shows the interfaces associated per VRf
# sh ip protocols vrf {name}                    - Shows the routing protocols configured in a VRF
# sh ip route vrf {name} [summary]              - Shows the VRF routing table
                                                - [Summary] Displays a summary of routes per VRF
# sh mpls forwarding vrf {name}                 - Shows labels allocated for the specified VRF
# sh ip cef vrf {name}                          - Shows per-VRF FIB table
# sh ip cef vrf {name} {ip-prefix} {detail}     - Shows details of an individual CEF entry, including label stack


#ip extcommunity-list {no} {permit|deny} rt {no}- Creates an extended community-list

#route-map {name} {permit|deny} [seq]
 #match ....                                    - Matches the necessary
 #set extcommunity rt {value} [additive]        - Attaches additional RTs in export-maps
                                                - [additive] Will append this RT and not overwrite original's set

#ip vrf {name}
 #ip vrf {vrf-name}                             - Creates a new VRF or enters configuration of an existing VRF
                                                - VRF names are case-sensitive
                                                - A VRF is not operational unless an RD is configured
 #rd {route-distinguisher}                      - This command assigns a route distinguisher to a VRF
                                                - The format can be ASN:NN or A.B.C.D:NN
 #route-target export {rt}                      - Specifies an RT to be attached to every route exported from this VRF to MP-BGP
 #route-target import {rt}                      - Specifies what MP-BGP routes to import into a VRF instance
 #import map {route-map}                        - Configures VRF import filtering
 #export map {route-map}                        - Configures selective VRF export
 #vpn id {oui:vpn-index}                        - (o) Configures a additional VPN identifier for a VRF
 #maximum routes {limit} [warn-thres|warn-only] - Configures the maximum number of routes accepted into a VRF table
                                                - [warn-threshold] Percentage value when a syslog message is logged
                                                - [warn-only] Creates a syslog error message when the maximum number of routes
                                                      exceeds the threshold
#interface fa0/0
 #ip vrf forwarding {name}                      - This command associates an interface with the specified VRF
                                                - This will clear the existing IP when configured

#router bgp 1
 #address-family ipv4 vrf {name}
  #neighbor {ip} maximum-prefix {limit} [threshold] [warning-only] [restart {interval}]
                                                - Controls how many prefixes can be received from a neighbor
                                                - [Threshold] Percentage value when a syslog message is logged (default = 75%)
                                                - [Warning-only] Warning when exceeding appose to dropping the session
                                                - [Restart] Re-establish the dropped session after the time specified
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
   PE to PE: MP-iBGP
*==============================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
         > Part7:MPLS Layer 3 VPNs
          > Configuring MPLS Layer 3 VPNs
           > Configuring the Core Network

- A protocol is required to carry VPNv4 routes between PE routers.
- RFC 2858 defines extensions to BGP-4 which enables it to carry multiple network layer protocols.
- The multi-protocol extensions are negotiated between BGP peers using an optional capabilities parameter in the BGP Open message.

- Multi-protocol extensions for BGP-4 define two new BGP optional transitive attributes used to advertise or withdraw routes.
- The attributes are MP_REACH_NLRI and MP_UNREACH_NLRI (Network Layer Reachability Information).
- The first two fields in these new attributes contain the AFI and the SAFI values.
- The AFI (Address Family Identifier) value identifies the network layer protocol.
- The SAFI (Subsequent Address Family Identifier) value identifies additional information about the type of NLRI carried.
- When the BGP peers exchange the multiprotocol extension capability, they also exchange AFI and SAFI numbers to identify what the
other
      BGP peer is capable of.

- AFI Values
 > AFI 1           - IPv4
 > AFI 2           - IPv6

- SAFI Values:
 > SAFI 1          - Unicast
 > SAFI 2          - Multicast
 > SAFI 3          - Unicast and Multicast
 > SAFI 4          - MPLS Label
 > SAFI 128        - MPLS Labeled VPN

- Multiprotocol extensions within BGP are implemented and configured as address-families (also known as contexts):
 > "address-family ipv4"             - Enters the IPv4 BGP context. Configuration relates to BGP in global table.
 > "address-family vpnv4"            - Enters the IPv4 MP-iBGP context. Configuration relates to BGP between PE routers.
 > "address-family ipv4 vrf name"    - Enters the per VRF MP-BGP context. Configuration relates to per VRF BGP tables.
                                     - MP-eBGP is also configured here, which is used for BGP communication between CE and PE routers.

- The exchange of addresses with BGP neighbors is enabled by default for the IPv4 address-family.
- If not required, this behavior can be disabled in two ways:
 > For all IPv4 routes               - "no bgp default ipv4-unicast"
 > For specific neighbor             - "no neighbor {ip} activate" (under the address-family ipv4)

- The address exchange for all other address-families is disabled be default.
- If required the specified neighbors within an address-family can be enabled with "neighbor activate"
```

```
CONFIG-SET : MP-BGP- Limit the Route-Exchange for Neighbors to Specific Address-Families
+------------------------------------------------------------------------------------------
|     router bgp 65000
|      neighbor 10.5.0.1 remote-as 65000                    - Specifies the BGP neighbors
|      neighbor 10.5.0.5 remote-as 65000
|      neighbor 10.5.0.9 remote-as 65000
|      !
|      no bgp default ipv4-unicast                          - Disables default IPv4 route exchange for all neighbors
|      address-family ipv4                                  - Enters the global BGP address-family
|       neighbor 10.5.0.1 activate                          - Manually enables IPv4 route exchange for these two neighbors
|       neighbor 10.5.0.5 activate                              because by default the IPv4-unicast was disabled
|      !
|      address-family vpnv4                                 - Enter PE-PE MP-BGP configuration context
|       neighbor 10.5.0.5 activate                          - Manually enables VPNv4 routes for these two neighbors
|       neighbor 10.5.0.9 activate
```

```
----------
 COMMANDS
----------
# sh ip bgp summary                             - Shows the global BGP neighbors and their session states
# sh ip bgp neighbors [ip]                      - Shows the global BGP neighbors and their negotiated protocols
# sh ip bgp vpnv4 all summary                   - Shows the VPNv4 BGP neighbors and their session states
# sh ip bgp vpnv4 [all|rd|vrf{name}] labels     - Shows the labels associated with VPNv4 routes
# sh ip bgp vpnv4 vrf {name}                    - Shows the per-VRF VPNv4 BGP table
# sh ip bgp vpnv4 all                           - Shows every VPNv4 BGP tables
# sh ip bgp vpnv4 rd {asn:nn}                   - Shows the VPNv4 routes matching the RD


#router bgp {asn}
 #neighbor {ip} remote-as {r-asn}               - All BGP neighbors must be configured under global BGP config
 #neighbor {ip} update-source loopback0         - MP-iBGP sessions should run between loopback interfaces
 #address-family vpnv4                          - Enters configuration of VPNv4 route exchanges for MP-iBGP sessions
  #neighbor {ip} activate                       - The BGP neighbor defined under BGP router configuration must be activated
                                                      for VPNv4 route exchange
  #neighbor {ip} next-hop-self                  - Changes the next-hop IP to the local peer address
  #neighbor {ip} send-community [std|ext|both]  - Extended communities are required for RT propagation
                                                - It's enabled by default when neighbor is activated for VPNv4. Always confirm!

 #no bgp default ipv4-unicast                   - Disables the default exchange of IPv4 routes for all neighbors
                                                - Neighbors that need to receive IPv4 routes must then be activated manually
 #no neighbor {ip} activate                     - Disables IPv4 route exchange on a per neighbor basis
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================================*
   PE to CE: Connected & Static Routes
*=========================================*
- The connected addresses, PE-to-CE address ranges, should be redistributed into MP-BGP to ensure connectivity.
- If MP-BGP advertise static routes within an MPLS VPN, they must be redistributed into MP-BGP on the configured PE router.
- A next-hop IP must be specified if a non-point-to-point interface is used.

----------
 COMMANDS
----------
#ip route vrf {name} {prefix} {mask} [interface] [next-hop] [global] [permanent] [tag {tag}]
                                                - Configures a per-VRF static route
                                                - [global]: The next-hop will be in the non-VRF global routing table
                                                - [permanent]: Route stays in the RIB even if interface is shut down

#router bgp {asn}
 #address-family ipv4 vrf {name}
  #redistribute static                          - Redistributes local VRF static routes into MP-BGP
  #redistribute connected                       - Redistributes connected VRF routes into MP-BGP



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   PE to CE: RIPv2
*=========================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
         > Part7:MPLS Layer 3 VPNs
          > Configuring MPLS Layer 3 VPNs
           > Configuring RIPv2 as the Routing Protocol Between the PE and CE Routers

- Only RIPv2 is supported for PE-CE routing.
- RIPv2 supports the use of address-families within the RIP process.
- The RIPv2 routes for a VRF should be redistributed into MP-BGP on the ingress PE router and back into EIGRP on the egress PE
router.


- For end-to-end RIP networks, the following applies:
 > On the ingress PE router, the RIP hop-count is copied into the BGP MED by default.
 > On the egress PE router, the RIP hop-count must be manually set for routes redistributed back into RIP, by one of the following
methods:
      >> Using a default metric for all RIP redistributed routes.
      >> Setting the hop-count in the "redistribute" command (this overwrites the default metric).
      >> Using the 'metric transparent' option to copy the BGP MED into the RIP hop-count (often for a consistent
            end-to-end RIP hop-count).
```

```
----------
 COMMANDS
----------
# sh ip route vrf {name} rip                     - Shows the RIP routes in the VRF-RIB table

#router rip
 #version 2                                      - Version 2 must be used
 #default-metric {hop-count}                     - Configures the default metric for redistributed routes
 #address-family ipv4 vrf {name}                 - Creates a per-VRF context within RIP routing process
  #redistribute bgp {asn} metric 5               - Redistributes BGP routes into RIP. Manually sets the RIP metric to 5
  #redistribute bgp {asn} metric transparent     - Redistributes BGP routes. Copies the BGP MED into the RIP hop-count

#router bgp {asn}
 #address-family ipv4 vrf {name}                 - Enters the MP-BGP VRF-context
  #redistribute rip                              - Redistributes RIP routes into MP-BGP. Rip hop-count is copied to BGP MED
  #no auto-summary                               - Disables auto-summarization
```

-=-

==========================

PE to CE: EIGRP

==========================

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
    > Part7:MPLS Layer 3 VPNs
     > Configuring MPLS Layer 3 VPNs
      > Configuring EIGRP as the Routing Protocol Between the PE and CE Routers

- EIGRP supports the use of address-families within the EIGRP process.
- The EIGRP routes for a VRF should be redistributed into MP-BGP on the ingress PE router and back into EIGRP on the egress PE router.
- The EIGRP AS-number for each VRF must be specified within each EIGRP VRF context.
- The metric of an EIGRP route is copied into the BGP MED attribute when redistributed into MP-BGP at ingress.
- When an MP-BGP route is redistributed into EIGRP, the BGP MED is NOT copied back to the EIGRP metric.
- The EIGRP metric must be manually set, otherwise the routes will not be advertised to the CE router:
  > Using a default metric for all EIGRP redistributed routes.
  > Setting the metric in the "redistribute" command (this overwrites the default metric).

- If the same EIGRP AS-number is used between VPN CE sites:
  > Internal EIGRP routes from one VPN site will be learned as internal EIGRP routes in other VPN sites.
  > External EIGRP routes from one VPN site will remain as external EIGRP routes.

- If the different AS-numbers are used between VPN CE sites:
  > Internal EIGRP routes from one VPN site will be learned as external EIGRP routes in other VPN sites.
  > External EIGRP routes from one VPN site will remain as external EIGRP routes.

```
- EIGRP SOO Loop-Prevention
 > DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing: EIGRP Configuration Guide, Release 12.4T
         > EIGRP MPLS VPN PE-CE Site of Origin

 > The SOO (Site-Of-Origin) BGP extended community can be used to prevent loops in dual-homed scenarios or when a
    backdoor link is configured between different VPN CE sites.
 > A unique SOO value must be configured for each VPN CE site.
 > When a router receives a route on an interface with a sitemap configured and the SOO of the route matches the configured SOO,
    the route is rejected.
 > This value should be used on the PE-CE interface.

----------
 COMMANDS
----------
# sh ip route vrf {name} eigrp                  - Shows the EIGRP routes in the VRF-RIB table

#router eigrp {pid}
 #address-family ipv4 vrf {name}                - Creates a per-VRF context within EIGRP routing process
  #autonomous-system {asn}                      - Configures the per-VRF AS-number
  #default-metric {b d l r m}                   - Configures the default metric for redistributed routes
  #redistribute bgp {asn} metric {b d l r m}    - Redistributes MP-BGP routes and sets the EIGRP composite metric
  #no auto-summary                              - Same as normal EIGRP, recommended to turn this off

#router bgp {asn}
 #address-family ipv4 vrf {name}                - Enters the MP-BGP VRF-context
  #redistribute eigrp {asn}                     - Redistributes EIGRP into MP-BGP

#route-map {mapname} permit {seq}               - Creates the site-map route-map
 #set extcommunity soo {xx:yy}                  - Specifies the SSO extended community
#interface s0/0
 #ip vrf sitemap {mapname}                      - Applies the SOO extended community attribute to routing updates
                                                     on the interface


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  PE to CE: OSPF
*==========================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
         > Part7:MPLS Layer 3 VPNs
          > Configuring MPLS Layer 3 VPNs
           > Configuring OSPF as the Routing Protocol Between the PE and CE Routers

- OSPF is not fully VPN-aware and does not support address-families. OSPF requires a separate OSPF process per VPN.
- The OSPF routes for a VRF should be redistributed into MP-BGP on the ingress PE router and back into OSPF on the egress PE router.
- The 'subnets' keyword is a requirement with OSPF otherwise only classful network addresses are redistributed.
```

- PE routers are seen as ABRs (Area Border Routers) and an MPLS VPN backbone is seen as a super Area 0.
- Several extended BGP communities were defined to carry OSPF route types and area types across an MPLS VPN backbone.
- The cost of an OSPF route is copied into the BGP MED attribute when the route is redistributed into MP-BGP at ingress.
- By default, OSPF to MP-BGP at PE redistributes intra-area and inter-area routes only. It does not redistribute external routes.
- Redistributing external routes must be explicitly specified using the "match internal external" command

- When an MP-BGP route is redistributed into OSPF the value of the MED is used to set the cost of the redistributed routes.
- This default behavior can be overwritten by:
  > Setting a default metric for all OSPF redistributed routes.
  > Setting the metric in the "redistribute" command (this overwrites the default metric).

- Routes redistributed from BGP into OSPF appear as inter-area summary routes or as external routes based on their original LSA type.
- Rules for MP-BGP routes redistributed into OSPF:
  > For original Type-1 or Type-2 LSAs, the redistributed routes will appear as inter-area summary LSA (Type-3).
  > For original Type-3 LSAs, the redistributed routes will appear as inter-area summary LSA (Type-3).
  > For Type-5 LSAs, the LSAs are re-originated as Type-5 LSAs with the egress PE as a ASBR.
  > For Type-7 LSAs, the LSAs are announced as Type-5 LSAs (as the route has already crossed area boundaries).
  > For non-original OSPF routes, normal BGP-OSPF redistribution rules apply (default LSA Type 5, route-type E2, and metric of 20).

- OSPF Domain-ID
  > Domain-ID is one of the new BGP extended communities used to reconstruct an original OSPF route at egress.
  > A domain-ID is used to indicate OSPF processes using different numbers (process-IDs) belonging to the same OSPF domain.
  > By default the domain-ID is set equal to the OSPF router process-ID.
  > If two PE routers use different OSPF process-IDs for the same VPN, the domain-ID should be manually set, otherwise all routes between the two VPN sites will appear as Type-5 external.

- OSPF Down-Bit
  > The down-bit is set in the options field of Type-3 LSA headers by egress PE routers to prevent loops (PE1-to-CE-to-PE2).
  > PE routers will never redistribute OSPF routes into MP-BGP if the down-bit is set.

- OSPF Domain-Tag
  > The domain-tag is set in the options field of Type-5 LSA headers by egress PE routers to prevent loops (PE1-to-CE-to-PE2).
  > The domain-tag is set to the configured value or the BGP AS-number is encoded into the domain-tag.
  > If a PE router receives an external route with the route tag matching its domain-tag, it will not redistribute the route into the MPLS VPN.

- Sham-Link
  > DOC-CD LOCATION
      > 12.4T Configuration Guides
        > Cisco IOS IP Routing: OSPF Configuration Guide, Release 12.4T
          > OSPF Sham-Link Support for MPLS VPN
  > Scenario when required
      >> When two sites belonging to the same area are interconnected via MPLS backbone and they have a backdoor link.
      >> The backdoor link will always be preferred, as OSPF prefers intra-area routes over inter-area routes.
  > A sham-link is a logical intra-area link across the MPLS backbone.
  > A separate /32 address space is required on each PE router for a sham-link.
  > This /32 must always be advertised by MP-iBGP, not by OSPF, and must belong to the VRF.
  > It is not possible to route traffic from one sham-link over another.
  > Sham-links get a default cost of 1 if the cost was not explicitly specified.

```
CONFIG-SET : MPLS OSPF Sham-Link between Two PEs (R1 and R2)
+---------------------------------------------------------------
|R1#  interface loopback1                                    - Creates a new loopback to use as sham-link end-point
|      ip vrf forwarding VPN                                 - Must be part of the client VRF
|      ip address 10.5.1.1 255.255.255.255                   - Can be any /32 IP address
|     !
|     router ospf 1 vrf VPN                                  - Enters the per-VRF OSPF process
|      area 0 sham-link 10.5.1.1 10.5.1.2 cost 3             - Creates the sham-link from 10.5.1.1 to 10.5.1.2
|      redistribute bgp 1 subnets                            - Usual MPLS OSPF config on a PE router
|      network 10.5.18.0 0.0.0.255 area 0                    - Enables OSPF on the PE-to-CE interface
|     router bgp 1
|      address-family ipv4 vrf VPN                           - Copy descretly owned by Kane  Bagwell
|       redistribute connected route-map loopback1           - This advertises Loopback1 to R2 via MP-BGP
|       redistribute ospf 1 vrf VPN match internal external - Usual MPLS OSPF config on a PE router
|
|R2#  interface loopback1
|      ip vrf forwarding VPN
|      ip address 10.5.1.2 255.255.255.255
|     !
|     router ospf 1 vrf VPN
|      area 0 sham-link 10.5.1.2 10.5.1.1 cost 3             - Creates the sham-link from 10.5.1.2 to 10.5.1.1
|      redistribute bgp 1 subnets                            - Usual MPLS OSPF config on a PE router
|      network 10.1.26.0 0.0.0.255 area 0                    - Enables OSPF on the PE-to-CE interface
|     router bgp 1
|      address-family ipv4 vrf VPN
|       redistribute connected route-map loopback1           - This advertises Loopback1 to R1 via MP-BGP
|       redistribute ospf 1 vrf VPN match internal external
>
>     %OSPF-5-ADJCHG: Process 1, Nbr 10.5.1.1 on OSPF_SL0 from LOADING to FULL, Loading Done
>                                                            - Shows the sham-link coming up
```

```
----------
 COMMANDS
----------
# sh ip route vrf {name} ospf                    - Shows the OSPF routes in the VRF-RIB table
# sh ip bgp vpnv4 vrf {name}                     - Shows the MP-BGP OSPF route and its ext-community values
# sh ip ospf sham-links                          - Shows the operational status and info of all sham-links

#router ospf {pid} vrf {name}                    - Starts a separate OSPF routing process for every VRF
 #domain-id ospf {domain-id}                     - Manually specifies the OSPF domain-ID instead of using the process-ID
 #domain-tag {value}                             - Manually specifies the value used in the OSPF domain-tag
 #default-metric {cost}                          - Configures the default metric for redistributed routes
 #redistribute bgp {asn} subnets                 - Redistributes MP-BGP routes into OSPF
                                                 - The subnets keyword is needed to avoid classful routes being from redistributed
 #area {id} sham-link {src-ip} {dst-ip} cost     - Configures a sham-link
#router bgp {asn}
 #address-family ipv4 vrf {name}                 - Enters the MP-BGP VRF-context
 #redistribute ospf {pid} [match [internal] [ex1] [ex2]]
                                                 - Without the OSPF match keyword specified, only internal OSPF routes
                                                       are redistributed into OSPF
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
  PE to CE: eBGP
*=========================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Multiprotocol Label Switching Configuration Guide, Release 12.4T
        > Part7:MPLS Layer 3 VPNs
         > Configuring MPLS Layer 3 VPNs
          > Configuring BGP as the Routing Protocol Between the PE and CE Routers

- To configure eBGP as a PE-CE routing protocol, a CE router is configured as an eBGP neighbor under the IPv4 VRF address-family.
- The neighbor must be activated under the address-family.

- AS-Override
 > AS-override is typically required if the same AS-number is used on different CE-sites interconnected by MP-BGP.
 > The command "as-override" circumvents the default BGP loop prevention by overriding the client AS-number.
 > If the first AS-number in the AS-path is equal to the neighboring CE site's AS-number it is replaced with the provider AS-number.
 > If the client AS-number was prepended multiple times, all AS-number occurrences are replaced with the provider AS-number.

- Allowas-in
 > By default, a BGP router cannot accept a prefix if the locally-configured AS-number is listed in the received AS-path list.
 > The command "allowas-in" circumvents the default BGP loop-prevention by ignoring the local AS-number in the AS-path list.

- SOO (Site-Of-Origin)
 > An extended community used to prevent PE-CE-PE and CE-PE-CE routing loops in multi-homed environments.
 > A route inserted into a VRF is not propagated to the CE router if the SOO attached of that route is equal to the SOO attribute
     associated with the CE router.

 ----------
 COMMANDS
 ----------
# sh ip route vrf {name} bgp                          - Shows the eBGP routes in a VRF-RIB table
# sh ip bgp vpnv4 vrf {name}                          - Shows the eBGP routes in the VRF MP-BGP table
# sh ip bgp vpnv4 vrf {name} summary                  - Shows the configured neighbor per VRF along with their status

#router bgp {asn}
 #address-family ipv4 vrf {name}                      - Configures/enters the MP-BGP VRF context
  #neighbor {ip|peer-group} remote-as {asn}           - Configures an eBGP neighbor in the VRF context, not in the global BGP config
  #neighbor {ip|peer-group} activate                  - eBGP neighbors must to be activated
  #neighbor {ip|peer-group} as-override               - PE router overrides client AS-number with its own AS-number.
  #neighbor {ip|peer-group} allowas-in {no}           - CE router allows its local AS-number to be listed in a received AS-path list
                                                      - {no} The number of times the local ASN can be listed from the LEFT

#route-map {mapname} permit {seq}                     - Configures SOO route-map
 #set extcommunity soo {xx:yy}                        - Specifies the SSO extended community
#router bgp {asn}
 #address-family ipv4 vrf {name}
  #ip vrf sitemap {mapname}                           - Applies a route map that sets SOO extended community attribute to
                                                            inbound routing updates received from this interface
```

VRF-Lite (Multi-VRF CE)

- DOC-CD LOCATION
  > 12.4T Configuration Guides
  > Cisco IOS IP Routing: OSPF Configuration Guide, Release 12.4T
  > OSPF Support for Multi-VRF on CE Routers

- VRF-lite allows the CE router the ability to maintain separate VRF tables, with the purpose of extending the privacy and security of an MPLS VPN down to the branch-office interfaces.
- The CE router separates traffic between client networks using VRF tables.
- The received route advertisements are inserted into the VRF routing table associated with the interface it arrived on.
- There is no MPLS functionality (LDP) on the CE router.
- Any routing protocol supported by normal VRF can be used in a multi-VRF CE implementation.

- OSPF Capability VRF-Lite
  > This feature disables the down-bit and domain-tag checks in OSPF.
  > The MPLS VPN PE routers advertise VPN routes with the down-bit set to CE routers.
  > Since a CE router acts as the PE router in VRF-lite, these checks should be disabled, because if a VRF-lite CE router receive routes with down-bit set it will discard them.
  > If a VRF-lite CE router connects to other OSPF routers, the "capability vrf-lite" should be configured under the OSPF process.

CONFIG-SET : VRF-lite CE Configuration Example (PE Config Remains Unchanged)

```
+---------------------------------------------------------------------------------
|      ip vrf BOB
|       description Friendly-Traffic
|      ip vrf BRUCE
|       description Unknown-Traffic
|      !
|      interface FastEthernet2/0.10
|       encapsulation dot1Q 10
|       ip vrf forwarding BOB                          - Places the interface into the BOB VRF
|       ip address 10.0.12.1 255.255.255.252
|      interface FastEthernet2/0.20
|       encapsulation dot1Q 20
|       ip vrf forwarding BRUCE                        - Places the interface into the BRUCE VRF
|       ip address 192.168.12.1 255.255.255.252
|      !
|      router ospf 1 vrf BOB
|       router-id 0.0.1.1
|        network 10.0.0.0 0.0.255.255 area 0
|        capability vrf-lite                           - Disables the down-bit and domain-tag checks in OSPF
|      router ospf 2 vrf BRUCE
|        router-id 0.0.1.2
|        network 192.168.0.0 0.0.255.255 area 0
|        capability vrf-lite                           - Disables the down-bit and domain-tag checks in OSPF
|
----------
 COMMANDS
----------
#router ospf {pid}
 #capability vrf-lite                            - Applies the multi-VRF capability to the OSPF process




*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
   Troubleshooting MPLS
*========================*

------------------------------------
   Troubleshooting MPLS and LDP
------------------------------------
- LDP Session Startup issues:
 > Do all the expected neighbors show up?                                            # sh mpls ldp discovery
 > Is MPLS enable on all the necessary interfaces?                                   # sh mpls interface
 > Are all the expected neighbors directly adjacent?                                 # trace {neighbor} (should be 1 hop)
      >> If not was a directed LDP session configured?                               # sh run | i target
      >> Is the non-adjacent neighbor reachable?                                     # ping {ip}
 > Are all the neighbors using the same protocol: LDP/TDP?                           # sh mpls interface detail
 > Any interface access-lists dropping ports 711 or 646?                             # sh ip interface | i line|list
 > Test connectivity between loopback interfaces                                     # ping {ip} source {ip}

- Labels are not being allocated:
 > Are labels allocated to local routes?                                             # sh mpls forwarding-table
 > Confirm CEF is enabled globally and on interfaces.                                # sh cef interface

- Labels are allocated, but not being distributed:
 > Does the adjacent LSR display the received labels?                                # sh mpls ldp bindings (on neighbor)
 > Is conditional label advertising configured?                                      # sh run | i advert

- Problems with large packets
 > Does an extended ping with packet sizes close to 1500 fail?                       # ping {ip} size 1500 df
 > Are the correct MTUs set?                                                         # sh mpls interfaces detail | i MTU
```

```
--------------------------------------
   Troubleshooting MPLS VPNs
--------------------------------------
       -----      -----      ---      -----      -----
      | CE1 |----| PE1 |----| P |----| PE2 |----| CE2 |
       -----      -----      ---      -----      -----


- Verifying proper routing information flow end-to-end (left-to-right):
 > Is CEF enabled on the ingress PE1 router interface and other needed ones?        PE1# sh cef interface
 > Are the CE routes received by an ingress PE1 router?                             PE1# sh ip route vrf {NAME}

 > Is there PE-to-PE connectivity?                                                  PE1# ping {pe2-lo0} source {lo0}
 > Are the MP-iBGP neighbor sessions established?                                   PE1# sh ip bgp vpnv4 * summary
 > Are VPNv4 routes propagated to other PE routers?                                 PE1# sh ip bgp vpnv4 all {prefix/length}

 > Do the routes redistributed into MP-BGP have proper extended communities?        PE1# sh ip bgp vpnv4 vrf {NAME} {prefix}
 > Have the PE routers exchanged their allocated VPN labels?                        PE1# sh ip bgp vpnv4 all labels
 > Has PE1 received a VPN label for the destination prefix from PE2?                PE1# sh ip bgp vpnv4 all labels | begin {prefix}
 > On PE1 does the BGP next-hop prefix have LDP label received from P?              PE1# sh mpls forwarding-table {bgp-nh}
 > Confirm the CEF entry is correct in the LFIB, with VPN and LDP label!            PE1# sh ip cef vrf {NAME} {prefix} detail
 > Is there an end-to-end (PE1-to-PE2) LSP? Verify labels on all LSRs:              ALL# sh mpls forwarding-table | in ^{label}

 > Is the BGP route selection process working correctly?                            PE2# sh ip bgp vpnv4 vrf {NAME} {prefix}
 > Are the expected best routes installed into the VRF-RIB on PE2?                  PE2# sh ip route vrf {NAME} {prefix}

 > If there are multiple equal cost links (CE1-to-PE1):
  -> Is only one route installed in the VRF-RIB on PE2?                             PE2# sh ip route vrf {NAME} {prefix}
  -> Is BGP multipath enabled?                                                      PE2# sh run | i maximum-paths.*address-fa


 > Are routes redistributed from BGP into the PE2-CE2 routing protocol?             PE2# sh ip route vrf {NAME} {prefix}
 > Are IPv4 routes propagated to CE2 routers?                                       CE2# sh ip route {prefix}
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY

```
 _______       _       _
|       |     | |  _  (_)                   _
| || || |_   _| |_| |_ _  ____ ____   ___ _| |_
| ||_|| | | | | |_   _| |/ ___|___ \ / __|_   _|
| |   | | | |_| | | | |_| | (___/ __) |__  | | |_
|_|   |_|____/|_|  \__)_|\____)_____(___/   \__)

*-=-=-=-=-=-=-=-=-=-=-=-*
|       INDEX           |
*-=-=-=-=-=-=-=-=-=-=-=-*
```

```
- Multicast Operation
- Addressing
      + Well-Known Reserved Ranges
      + Well-Known Reserved Addresses
      + Multicast MAC Addressing
- IGMP (Internet Group Management Protocol)
      + Queries and Reports
      + IGMPv2
      + IGMPv3
      + IGMP Snooping
- PIM (Protocol-Independent Multicast)
      + DR (Designated Router)
      + AF (Assert Forwarder)
      + PIM-DM (Dense Mode)
      + PIM-SM (Sparse mode)
      + PIM-BIDIR (Bi-Directional)
      + Tree-Types
      + Shortest Path Switchover (SPT/RPT)
- RPF (Reverse Path Forwarding)
      + Static Mroute
- RP Assignments
      + Static
            o Override
      + Auto-RP
            o Sparse-Dense
            o Auto-RP Listener
      + BSR (Bootstrap Router)
      + Anycast RP
- NBMA Mode
- Multicast Over GRE
- Multicast Stub Routing
- Filtering
      + Static RP
            o Filtering Specific Groups
      + Auto-RP Filtering
            o RP Group Filtering
            o MA Filtering c-RPs
      + Multicast Boundary
      + BSR Border Filtering
      + PIM Neighbor Filtering
```

  - + Group-to-RP Filtering
  - + IGMPz Join Filtering
  - + Multicast Route-Limit
  - + Multicast Rate-Limit
- Multicast Scoping
  - + TTL Scoping
  - + Administrative Scoping
- Additional Multicast Features
  - + Multicast Helper
  - + SDR Listener Support
  - + Load Splitting Multicast Traffic
  - + Multicast Heartbeat
- SSM (Source Specific Multicast)
- MSDP (Multicast Source Distribution Protocol)
- PGM (Pragmatic General Multicast)
- MRM (Multicast Routing Monitor)
- MVR (Multicast VLAN Registration)
- DVMRP (Distance Vector Multicast Routing Protocol)
- Troubleshooting Multicast
  - + General Troubleshooting
  - + Sparse Mode
  - + RPF Failures
  - + RPF Failure Solutions

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
```

Multicast Operation

```
*=========================*
```

- DOC-CD LOCATION
  - > 12.4T Configuration Guides
    - > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
      - > Configuring Basic IP Multicast

- Multicast is UDP-based. Since the protocol is unreliable by design, error checking is the application's responsibility.
- A multicast server is usually the source of a multicast feed and the clients are usually the destination.
- A source address can never be a multicast address; it is always a unicast address.

```
-----------
 COMMANDS
-----------
# sh ip mroute                                  - Shows the multicast routing table
# sh ip multicast interface [int]               - Shows multicast details for the interface
# clear ip mroute *                             - Clears routes from the multicast routing table

#ip multicast-routing                           - Globally enables multicast routing on a routers
#ip multicast-routing distributed               - Globally enables multicast routing on a 3560
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================*
    Addressing
*=====================*
- Multicast address range formats:
      >> Range          -     224.0.0.0 - 239.255.255.255
      >> Prefix         -     224.0.0.0/4
      >> Subnet mask    -     224.0.0.0 240.0.0.0
      >> Inverse mask   -     224.0.0.0 15.255.255.255

- Well-Known Reserved Ranges
 > Reserved Link Local Address Range                                224.0.0.0/24
      >> These are non-routed addresses used only on a local link (TTL=1).


 > Reserved Local Routed Address Range                              224.0.1.0/24
      >> Reserved for local network protocols.


 > Reserved SSM (Source-Specific Multicast) Range                   232.0.0.0/8
      >> Allows IGMPv3 host applications to select the source for a multicast group.
      >> SSM makes multicast routing more efficient.


 > Reserved GLOB Address Range                                      233.0.0.0/8
      >> Meant to be used by registered ASN owners for global uniqueness.
      >> The 2nd and 3rd octet gets mapped the unique ASN.
      >> The 4th octet is used for internal purposes.


 > Reserved Private Multicast Address Range                         239.0.0.0/8
      >> Administratively scoped address range.
      >> Private internal usage to a network ONLY.


- Well-Known Reserved Multicast Addresses
 > 224.0.0.1             - All multicast hosts
 > 224.0.0.2             - All multicast routers
 > 224.0.0.4             - DVMRP routers
 > 224.0.0.5             - OSPF routers
 > 224.0.0.6             - OSPF DR routers
 > 224.0.0.9             - RIPv2 routers
 > 224.0.0.10            - EIGRP routers
 > 224.0.0.13            - PIM routers
 > 224.0.0.22            - IGMPv3
 > 224.0.0.25            - RGMP
 > 224.0.1.39            - Auto-RP Announce (RP)
 > 224.0.1.40            - Auto-RP Discovery (MA)
```

```
- Multicast MAC addressing
 > Assigning a layer3 multicast address to a multicast group/application automatically generates a layer2 multicast MAC address.
 > The MAC is formed as follow:
      >> Always starts with 0100.5E.
      >> Followed by a binary 0.
      >> Followed by the last 23 bits of the multicast IP address converted to HEX.

 > Example
      >> Multicast IP: 231.205.98.177 = 01-00-5E-4D-62-B1.

      >> Take the IP into binary: 11100111.11001101.01100010.10110001.

      -> Convert the last 23 bits in HEX:
          11100111.(0)100 1101.0110 0010.1011 0001
                    \   /\   /\   /\   /\   /\   /
                     \ /  \ /  \ /  \ /  \ /  \ /
          01-00-5E-   4    D -  6    2 -  B    1

      Combine the last output to get the multicast MAC address of 01-00-5E-4D-62-B1
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=================================================*
    IGMP (Internet Group Management Protocol)
*=================================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Customizing IGMP

- IGMPv1 and v2 use protocol number 2, with an IP TTL of 1.
- IMGP is enabled automatically when PIM is enabled.
- Designed to enable communication between a multicast router and connected hosts.
- A host informs the local multicast router of its need to receive traffic for a specific multicast group.
- Or a host informs the local multicast router that it wants to leave a multicast group.

- Multicast routers use IGMP to track what multicast groups should be forwarded on which interfaces.
- When joining a group/launching an application the multicast MAC is calculated and the host's NIC will start listening for that
      multicast MAC address too.

- Host Membership Query:
 > Routers use queries to discover the presence of multicast group members on a subnet.
 > A general membership query is sent to the group address 0.0.0.0.
 > A group-specific query is sent to the group address that is queried.

- Host Membership Reports:
 > Hosts use reports in reply to queries or
 > To inform the router of their desire to receive multicast traffic.
```

```
- IGMPv2 includes the following features compared to IGMPv1:
 > Leave-Group Messages
      >> Used by hosts to notify the router that they want to leave the group.
      >> Sent to destination 224.0.0.2.
 > Group-Specific Query Messages
      >> Allow the router to send a query for a specific group instead of ALL groups.
 > Maximum response time (MRT)
      >> The time a host has to respond to a query with a report.
 > Querier Election Process
      >> Selects the preferred router to send query messages on a segment with multiple routers.
      >> The router with the lowest IP address is elected as the IGMP querier.

- IGMPv2 Timers
 > Query interval
      >> The time period between general queries sent by a router.
 > Query response interval
      >> The maximum response time for hosts to respond to the periodic general queries.
 > Group membership interval
      >> The time period after which, if a router does not receive an IGMP report, the router
         concludes that there are no more members of the group on the subnet.
 > Other querier present
      >> The time period after which, if the IGMPv2 non-querier routers do not receive an IGMP query from
         the querier router, the non-querier routers conclude that the querier is dead.
 > Last member query interval
      >> The maximum response time inserted by IGMPv2 routers into the group-specific queries and the time
         period between two consecutive group-specific queries sent for the same group.

- IGMPv3
 > Allows a host to filter incoming traffic based on the source IP addresses from which it is willing to receive packets
 > This feature is called SSM (Source-Specific Multicast).
 > SSM uses the range 232.0.0.0/8.
 > IGMPv3 leave group messages are sent to destination 224.0.0.22.

- IGMP Snooping
 > IGMP snooping enables a switch's software to eavesdrop on the IGMP conversations between multicast hosts and the router.
 > The switch examines IGMP messages and learns the port locations of multicast routers and group members.

-----------
 COMMANDS
-----------
# ping {multicast-ip}                           - Emulates a server multicasting to a group
# sh ip igmp group                              - Shows IGMP group membership information
# sh ip igmp interfaces                         - Shows IGMP interface information

#interface fa0/0
 #ip igmp join-group {m-ip}                     - Allows an interface to emulate a client config by joining a multicast group
                                                - Interface will process multicast traffic, so it will respond to pings
 #ip igmp static-group {m-ip}                   - Only emulates the join, but doesn't process multicast traffic
                                                - Interface will not respond pings
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================================*
    PIM (Protocol-Independent Multicast)
*=========================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring Basic IP Multicast

- PIM is used for router to router communication, thus PIM is a multicast routing protocol.
- PIM does not advertise its own topology information, instead it relies on unicast routing protocols.
- Multicast groups can be either sparse OR dense.
- Multicast interfaces can be configured as either/both.
- PIMv2 sends hello messages every 30 sec on PIM enabled interfaces.
- PIMv2 uses a holdtime value of 3x the hello interval.
- PIMv2 uses protocol 103 and the reserved multicast address 224.0.0.13 (all-PIM-routers).
- The PIM mode only determines how traffic is sent from an interface, not how it is received.

- DR (Designated Router)
 > A DR is necessary for each multi-access LAN running IGMP, to allow a single router to send IGMP host-query messages to solicit
      host group membership.
 > The highest IP on a LAN segment will be elected as the DR through the election process.

- AF (Assert Forwarder)
 > On multi-access networks one router will be elected and responsible for forwarding multicast traffic.
 > The assert election criteria:
      1- Admin distance to the source.
      2- Lowest metric to the source.
      3- If there is a tie, then the router with the highest IP address will be elected.

*----------------------*
  PIM-DM (Dense Mode)
*----------------------*
 > Designed for networks where many multicast clients are tightly spaced together in small networks.
 > When a PIM-DM router receives a multicast packet, it first performs an RPF check to the multicast source.
 > If the RPF check succeeds, the router forwards a copy of the packet out of all interfaces except the following:
      >> The interface the on which the packet was received
      >> Interfaces where a prune message was received from downstream routers stating that they do not want that traffic.

 > Typically uses Source-Based-Tree or Shortest-Path-Tree (SPT) as the multicast source is the ROOT of a tree.
 > With Dense mode the SPT may differ for each combination of source and multicast group.
 > The notation (S,G) refers to a particular dense SPT.
 > The SPT includes all interfaces by default but the PIM Prune message allows interfaces to be removed.

 > Dense mode uses implicit joins, by assuming all traffic is wanted by all clients, unless specified otherwise.
 > Flood and Prune behavior:
      >> Flood - All clients are assumed to be members of all multicast groups.
      >> Prune - This instructs upstream routers to stop sending the traffic for the particular group.
 > The term OIL- 'Outgoing Interface List' refers to the list of interfaces in a forwarding state,
      listing entries from a router's multicast routing table.
 > A pruned state on the outgoing interface list is indicated as NULL.
 > A multicast router can have one or more interfaces in the OIL, but only one interface is allowed in the
      incoming interface list.
 > An incoming interface of 'RPF nbr of 0.0.0.0' indicates that the connected device is the source for the group.
 > An incoming interface of 'Null,RPF neighbor 0.0.0.0' indicates that the source is still unknown.
 > Upstream is towards the source and downstream is towards the multicast group/hosts.
 > A router can send a graft message to an upstream neighbor, to which it had formerly sent a prune message
      asking the upstream router to put that link back into the forwarding state for a particular (S,G) SPT.


*-----------------------*
  PIM-SM (Sparse mode)
*-----------------------*
 > Designed for larger than PIM-DM networks where only a few clients exist.
 > Sparse-mode protocols do not forward multicast group traffic out of any interface, unless a router requests packets to be
      sent to a particular multicast group.
 > PIM-SM works efficiently with a relatively small number of multicast senders, but runs into issues with many senders.
 > A downstream router will request this either because another downstream router requested it or because a directly
      connected host sent an IGMP join message for that group.
 > Sparse mode uses explicit joins whereby no traffic is sent unless requested.
 > Sparse mode employs a RP (Rendezvous Point) to process join requests.


 > Order of operation with sparse mode:
      >> A multicast source begins sending multicast traffic to a group, e.g. 226.1.1.1.
      >> The connected router sends a unicast PIM register messages to the defined RP (10.5.1.1,226.1.1.1)
      >> Transit PIM routers will not show this (S,G) since it is a unicast PIM message.
      >> The RP will ignore this multicast traffic and respond with a register 'stop messages', until it receives
            a PIM join for that group.
      >> A host somewhere then sends an IGMP JOIN (*,226.1.1.1) to its gateway router.
      >> This downstream gateway router then sends a PIM-SM JOIN for (*,226.1.1.1) to the defined RP.
      >> PIM routers in the transit path will install (*,226.1.1.1) into their multicast tables.
      >> Once the RP receives the join, it will start forwarding traffic sent for 226.1.1.1 to this downstream router.
      >> When the downstream router no longer wants multicast traffic, it sends a PIM-SM Prune (10.5.1.1,226.1.1.1) to the RP.


 > Sparse mode typically uses a Shared-Path-Tree or Root-Path-Tree (RPT), because it is rooted at the RP.
 > Sparse mode initially causes multicasts to be delivered in a two-step process:
      1- Packets are sent from the source to the RP.
      2- The RP forwards the packets to the subnets that have hosts who requested copies of those multicasts.
            A shared tree is used for the second part of the process.

 > An incoming interface of 'Null, RPF neighbor 0.0.0.0' indicates this router is the RP.
 > An outgoing interface of 'Null' indicates the RP does not know of any clients.
```

```
*------------------------------*
  PIM-BIDIR (Bi-Directional)
*------------------------------*
 > DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring Basic IP Multicast
         > Configuring Bidirectional PIM

 > PIM-SM uses a SPT (Source Path Tree) between the multicast source and the RP (S,G) and a RPT (Root Path Tree/Shared tree) to
      manage the forwarding of multicast traffic from the RP to receivers (*,G).
 > The problem is that number of SPTs grows when the number of multicast sources increase, which makes PIM-SM less efficient in
      networks with large number of multicast senders.
 > PIM-BIDIR is closely related to PIM-SM and arguably addresses the PIM-SM scalability problem by only building RPTs.
 > With PIM-BIDIR a multicast source forwards traffic downstream towards the RP in a (*,G) fashion, while the RP use the
       same RPT to forward multicast traffic upstream towards the receiver (*,G).
 > This allows PIM-BIDIR to scale better than PIM-SM.
 > Since there is no SPT, a SPT switchover is not possible and traffic will always be forwarded through the RP.

 > In PIM-BIDIR, traffic is allowed to be passed up the RPT toward the RP. (no RPF check)
 > But to avoid multicast packets from looping, PIM-BIDIR introduces a new mechanism called the DF (Designated Forwarder)
      election, which establishes a loop-free tree rooted at the RP.

 > DF-Election
      >> On every network segment, all PIM routers selects one router as the DF for every RP of bidirectional groups.
      >> The DF-elected router is responsible for forwarding multicast packets received on that network segment
            upstream to the RP.
      >> The election of a DF is based on the best unicast route to the RP.
      >> This ensures that only one copy of every multicast packet will be sent to the RP.

 > PIM-SM and PIM-BIDIR can coexist using the same RPs.
 > If PIM-BIDIR is required in a multicast domain, all routers including the RP must be configured to support it.
 > The RP must be configured operate in PIM-BIDIR mode, else if the keyword 'bidir' is omitted, groups specified will
      default to PIM-SM.
 > If 'bidir' is configured, a bidirectional RP advertises all groups as bidirectional by default. An ACL on the RP can be
      used to specify a list of groups to be advertised as bidirectional.
```

```
CONFIG-SET : Using PIM-BIDIR, PIM-SM and PIM-DM together
+-----------------------------------------------------------
|Example 224/8 includes bidirectional groups, 225/8 is dense mode and 226/8 is sparse mode.
|
|RP#
|     ip multicast-routing                                   - Enables multicast routing
|     ip pim bidir-enable                                    - Enables PIM-BIDIR
|     !
|     access-list 45 permit 224.0.0.0 0.255.255.255          - Matches groups that operate in PIM-BIDIR
|     access-list 45 deny   225.0.0.0 0.255.255.255          - Groups with the 'deny' keyword will operate in dense mode
|     access-list 46 permit 226.0.0.0 0.255.255.255          - Matches groups that operate in PIM-SM
|     !
|     ip pim send-rp-announce lo0 scope 10 group 45 bidir  - Enables PIM-BIDIR group support on the RP for ACL-45
|     ip pim send-rp-announce lo1 scope 10 group-list 46   - Enables PIM-SM group support on the RP for ACL-46
|     ip pim send-rp-discovery lo0 scope 10
|     !
|     interface loopback0                                  - Configures loopback used for the PIM-BIDIR RP
|      ip address 10.5.224.1 255.255.255.255
|      ip pim sparse-mode
|     !
|     interface loopback1                                  - Configures loopback used for the PIM-SM RP
|      ip address 10.5.226.1 255.255.255.255
|      ip pim sparse-mode
|     !
|      interface fastethernet0/0
|      ip address 10.5.2.5 255.255.255.0
|      ip pim sparse-mode
|
|
|Other routers#                                            - Configuration the same as usual on other multicast routers
|     ip multicast-routing
|     ip pim bidir-enable
|     !
|     interface fastethernet0/0
|      ip address 10.5.2.4 255.255.255.0
|      ip pim sparse-mode
|
```

```
*---------------*
  Tree-Types
*---------------*
- 2 Types:
 > Source-Tree or SPT (Shortest-Path-Tree)
      >> Has the multicast source as the ROOT of a tree.
      >> A SPT tree is built using the least cost route between the source and the destination. This is the default tree type.
 > Shared-Tree or RPT (Root-Path-Tree)
      >> Is rooted at the RP.
      >> With RPT all multicast packets are sent to the RP and then down to the receivers.

- The RPF check is performed differently based on the tree type:
 > Using SPT, the RPF check is done against the source of the multicast traffic.
 > Using RPT, the RPF check is done against the RP and not against the source of the multicast traffic.

- The default type is SPT (source-tree).
 > Changing the tree type from a source-tree to a shared-tree could be used as a workaround with an RPF failure,
      specifically when the use of static mroutes (multicast routes) or changing of the unicast routing is not allowed.
```

```
*-----------------------------*
  Shortest Path Switchover
*-----------------------------*
 > Shortest path switchover means calculating and changing to the most efficient path. This happens by default.
 > Once a destination DR receives (S,G) feed it may choose to switch to a SPT by sending a new (*,G) PIM JOIN towards the
     source (S, instead of towards the RP.
 > This is indicated in the mroute table as 'T' when the SPT bit is set.
 > You can disable this default behaviour and force the traffic to pass through the RP by using the command
  "ip pim spt-threshold".


-----------
 COMMANDS
-----------
# mrinfo                                        - Shows info about PIM neighbor connectivity
# sh ip pim interface                           - Shows the interfaces with PIM configured
                                                - Shows the mode, query interval and DR per segment
# sh ip pim rp mapping                          - Shows the PIM group-to-RP mappings
# sh ip pim neighbors                           - Shows PIM neighbors
# sh ip pim int df                              - Shows info about the elected DF for each RP of an interface


#ip multicast-routing                           - Globally enables multicast routing on a routers

#ip pim bidir-enable                            - Enables PIM-BIDIR support globally

#interface eth0
 #ip pim dense-mode                             - Enables PIM-DM
 #ip pim sparse-mode                            - Enables PIM-SM

#ip pim spt threshold {infinity | kbps} [acl]   - Disables the SPT switchover for all groups or specific groups
                                                - {kbps} Traffic rate in kilobits p/sec before a switchover is initiated
                                                - {infinity} Never switch to source-tree

                                                >>> PIM-BIDIR RP configurations options <<<
#ip pim rp-address {ip} [group] [override] bidir
                                                - Configures a static RP and specifies PIM-BIDIR

#ip pim rp-candidate {int} [group-list {acl} interval {sec}] bidir
                                                - Configures the BSR c-RP and specifies PIM-BIDIR

#ip pim send-rp-announce {src-int} scope {ttl} [group-list {acl} interval {sec}] bidir
                                                - Configures a c-RP for auto-RP
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
   RPF (Reverse Path Forwarding)
*==================================*
- PIM relies on unicast routing protocols for the RPF check.
- The definition of a RPF check from CCO:
 > When a router receives a multicast packet, the source IP is taken and used to determine the reverse path interface.
 > The unicast routing table is used to determine the interface used to forward traffic back to this source IP.
 > If this reverse path interface matches the interface the multicast traffic was received on, the RPF check is successful.
 > If the RPF check is successful the packet is forwarded- if not, the packet is dropped.

- The RPF check basically verifies that the incoming interface for a multicast feed is the outgoing interface for unicast
       traffic back towards the source.

- Static "mroute" overrides unicast information by allowing non-RPF interfaces to receive multicast traffic.
- Static "mroute" has no influence on data flow. It is used only for the RFP check.

- To verify RPF failure examine the output from these commands:
 > "show ip mroute count"          - The 'RPF failed' counter indicates any RFP failures
 > "debug ip mpacket"              - A message of 'not RPF interface' usually points to a RPF failure
                                   - A message of 'mforward' means multicast traffic is being forwarded

!!NOTE!! The command "no ip mroute-cache" under the interfaces is required to debug transit multicast traffic.


-----------
 COMMANDS
-----------
# show ip rpf {ip}                              - Shows the RPF information
# sh ip mroute                                  - Shows the multicast routing table
# sh ip mroute active                           - Shows the active multicast traffic
# sh ip mroute count                            - Shows multicast routing statistics
                                                - "RPF failed" counter will point out RFP failures
                                                - "Other Drops" could indicate a lack of client requests

# debug ip pim                                  - Shows the PIM events and transactions
# debug ip mpacket                              - Shows the packet information of process switched transit traffic
                                                - Requires "no ip mroute-cache" on transit interfaces


#ip mroute {source ip} {mask} {nh}              - Changes the interfaces for which an incoming multicast feed is expected on
                                                - {NH} Unicast next-hop could be IP or interface. For NBMA must be a IP

#ip mroute 0.0.0.0 0.0.0.0 {nh}                 - Applies to any source
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
    RP Assignments
*======================*
- RP assignments can be:
 > Static
      >> Static RP assignments by default are LESS preferred than dynamically learned RPs.
 > Dynamic
      >> Cisco proprietary Auto-RP or
      >> Standards based BSR (Bootstrap Router).

- To use redundant RPs, Cisco offers two methods:
 > Anycast RP using the MSDP (Multicast Source Discovery Protocol).
 > BSR (Bootstrap Router).


*-------------------*
  Auto-RP
*-------------------*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring Basic IP Multicast
         > Configuring Sparse Mode with Auto-RP

- Cisco proprietary protocol.
- Steps used by auto-RP to determine the RP:
 > Each c-RP (candidate-RP) configured to use auto-RP will announce itself and its supported multicast groups via
      RP-announce messages (224.0.1.39).
 > The auto-RP MA (Mapping Agent), which may or may not be the RP router, gathers information about all c-RPs
      by listening to the RP-announce messages.
 > The MA builds a mapping table listing all the RPs and their received announcements.
 > The MA then picks the RP with the highest IP address if multiple RPs support the same multicast groups.
 > The MA sends RP-discover messages (224.0.1.40) advertising the mappings and RPs to other routers.
 > All multicast routers listen for packets sent to 224.0.1.40 to learn the mapping information and find the correct RP to use
      for each multicast group.
 > Auto-RP announcements are subject to a RPF CHECK!!
 > Auto-RP is configured on the:
      >> C-RP (224.0.1.39) with "ip pim send-rp-announce".
      >> MA (224.0.1.40) with "ip pim send-rp-discovery".


- PIM must be enabled on the loopback interface if used for discovery or advertisement.
- There is design problem with auto-RP. It was designed for sparse mode but to find the mappings, dense mode behavior is needed.
- The problem with auto-RP router in sparse mode is:
      >> A router can't join the auto-RP groups without knowing the RPs address.
      >> A router doesn't know where the RP is without joining the auto-RP groups.
```

- The c-RP announcements and MA discovery messages require dense mode, so there are two workarounds:
    - >> Sparse-Dense mode
        - + With this workaround, dense mode is used for groups without RP (including: 224.0.1.39/224.0.1.40).
        - + And sparse mode is used for all other groups.
        - + For every group where the RP is unknown, the tree will fall-back to dense mode.
        - + Configured on all transit interfaces with "sparse-dense-mode".
    - >> Auto-RP Listener
        - + With this workaround ONLY 224.0.1.39 and 224.0.1.40 run in dense mode.
        - + Sparse-mode is used for all other groups.
        - + The interfaces are required to be in dense mode.
        - + Configured on all routers with "ip autorp listener".
        - + Typically used when there are only sparse mode transit interfaces.

BSR (Bootstrap Router)

- DOC-CD LOCATION
    - > 12.4T Configuration Guides
        - > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
            - > Configuring Basic IP Multicast
                - > Configuring Sparse Mode with a Bootstrap Router

- BSR works similarly to auto-RP.
- BSR is often referred to as PIMv2. !!Watch out for this terminology!!
- One router acts as the BSR, which is similar to the MA in auto-RP.
- The BSR receives mapping information from the RPs and then advertises the information to other routers.
- However there are differences between the BSR and the auto-RP MA:
    - > The BSR router does not pick the best RP for each multicast group; instead, the BSR router sends all group-to-RP mapping information to the other multicast routers inside PIM messages.
    - > The BSR floods the mapping information in a bootstrap message sent to the all-PIM-routers multicast address (224.0.0.13).
    - > PIM routers each independently pick the best RP for each multicast group by running the same hash algorithm on the information in the bootstrap messages.
    - > The flooding of bootstrap messages does not require the routers to have a known RP or to support dense mode.

- PIM-SM routers flood bootstrap messages out of all non-RPF interfaces, downstream/away from the BSR, which in effect guarantees that at least one copy of the message makes it to every router.
- With BSR, c-RPs can make use of a priority value to give preference to one RP over another.
- The highest priority value is preferred and default equals 0.

- When multiple c-RPs are advertising the same group-addresses, the Cisco IOS will make its decision based on the c-RP that advertises the longest match in the announced groups, before considering the RP-priority.
- BSR supports redundant RPs and redundant BSRs.
- Multiple BSR routers can be configured.
    - > Each candidate BSR (c-BSR) router sends bootstrap messages which include the priority of the BSR router and its IP address.
    - > The highest-priority BSR wins, or if a tie occurs, the highest BSR IP address wins.
    - > The winning BSR, or the preferred BSR, will continue to send bootstrap messages, while the other BSRs listen.
    - > If the preferred BSR’s bootstrap messages stop the redundant BSRs will attempt to take over.

```
*-------------------*
   Anycast RP
*-------------------*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring Basic IP Multicast
         > Configuring Sparse Mode with Anycast RP

- The key differences between using anycast RP or either auto-RP or BSR relate to how the redundant RPs are used.
- With anycast RP-to-RP redundancy load sharing can be achieved with multiple RPs concurrently acting as
      the RP for the same group.
- Without anycast RP-to-RP redundancy, only one router is allowed to be the active RP for each multicast group. Load sharing of
      the collective work of the RPs can be accomplished by using one RP for some groups and another RP for other groups.
- The way anycast RP works is to have each RP use the same IP address. The RPs must advertise this address,
      typically as a /32 prefix, with IGPs.
- At the end of the process, any packets sent to the RP are routed as per the IGP, to the closest RP.
- The two biggest benefits with anycast RP are as follows:
 > Multiple RPs share the load for a single multicast group.
 > Recovery after a failed RP happens quickly. If an RP fails, multicast traffic is only interrupted for the
       time it takes the IGP to converge to point to the other RP sharing the same IP address.


-----------
 COMMANDS
-----------
# mrinfo                                   - Shows information about PIM neighbor connectivity
# sh ip pim interface                      - Shows the interfaces with PIM configured
                                           - Shows the mode, query interval and DR per segment
# sh ip pim rp mapping                     - Shows the PIM group-to-RP mappings
# sh ip pim bsr-router                     - Shows the BSR router and its information

#ip pim rp-address {ip} [acl] [override]   - Statically configures the RP on all routers including the RP
                                           - [acl] Limits the groups a RP will advertise via PIM-JOIN messages
                                           - [override] Overrides dynamically learnt RP mappings

#ip pim send-rp-announce {src-int} scope {ttl}  [group-list {acl} interval {sec}]
                                           - Configures a c-RP for auto-RP
                                           - {int} The IP address to advertise as the c-RP
                                           - {ttl} The TTL of the advertisement messages
                                           - {acl} See filtering section below
                                           - {interval} How often the candidate announcements are sent

#ip pim send-rp-discovery {src-int} scope {ttl} - Configures the auto-RP MA
                                           - {int} The IP address to advertise as the MA
                                           - {ttl} Is the TTL of the discovery messages

#no ip pim dm-fallback                     - Disables the tree to fall-back to dense mode if RP is unknown
#interface fa0/0
 #ip pim sparse-dense-mode                 - Uses sparse mode if the RP is known else dense mode is used
                                           - Alternative to this is "ip autorp listener"
```

```
#ip pim autorp listener                        - Uses dense mode for 224.0.1.39 and 224.0.1.40 only
                                               - Alternative to using sparse-dense
#ip pim bsr-candidate {int} [priority]         - Defines the BSR(s)
#ip pim rp-candidate {int} [group-list {acl} interval {sec}]
                                               - Configures the BSR c-RP
```

-=-

======================

NBMA Mode

======================

- DOC-CD LOCATION

```
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring Basic IP Multicast
         > Configuring IP Multicast over ATM Point-to-Multipoint VCs
```

- Reason for NBMA mode:
  - > The incoming interface cannot be the same as the outgoing interface in the multicast routing table.
  - > This creates a problem with hub/spoke environments.
- NBMA mode allows sparse groups to list a remote IP address instead of the interface in the multicast routing table.
- NBMA mode prevents unnecessary forwarding to all spokes and allows spoke-to-spoke multicast communication.
- Sparse-mode requires NBMA mode to allow traffic to enter and exit the same interface.
- Do not use PIM NBMA mode on multicast-capable ethernet LANs.

COMMANDS

```
#interface s0/0
 #ip pim spare-mode                            - NBMA mode only works with sparse mode interfaces
 #ip pim sparse-dense mode                     - This command will produce an error, just ignore it
 #ip pim nbma-mode                             - Enables NBMA mode on the interface
```

-=-

======================

Multicast Over GRE

======================

- DOC-CD LOCATION

```
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring Basic IP Multicast
         > Tunneling to Connect Non-IP Multicast Areas
```

- Multicast over GRE is necessary when multicast traffic must be sent over an area where IP multicast is not supported.
- When using tunnels to transport multicast traffic, RPF failures are common. This can be fixed with a static mroute.
- A GRE tunnel by default does not maintain state. If only one end is configured it will show UP/UP, even before the other end is configured.
- To get around this enable keepalive support with the "keepalive" command under tunnel interface.
- Always make sure the source and destination IP's are not routed through the tunnel interface.

```
-----------
 COMMANDS
-----------
#interface tunnel 0
 #ip unnumbered loopback0
 #tunnel source y.y.y.y                          - Must be same as destination address on the other side
 #tunnel destination x.x.x.x                     - Must be same as source address on the other side
 #keepalive {sec} {retries}                      - Manage the tunnel state
 #ip pim dense-mode                              - Enables PIM over the tunnel
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*===========================*
  Multicast Stub Routing
*===========================*
- Used to prevent periodic flood and prune behavior over low-bandwidth links.
- Stub routing prevents any PIM neighbor relationship on segment A while retaining multicast connectivity behind segment A.
- Multicast traffic is still passed to the end hosts behind the multicast router in segment A.
- Remote clients forward IGMP join requests to a stub router that forwards them onto a central router.
- Conceptually similar to DHCP relay.

-----------
 COMMANDS
-----------
 #ip pim neighbor-filter {acl}                   - On the central router, filters all PIM messages based on the ACL
                                                 - Prevents PIM adjacencies from forming
 #ip igmp helper-address {central-router-ip}     - On the stub router, forwards all IGMP messages to central router
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*======================*
  Filtering
*======================*
- Static RP Filtering

CONFIG-SET: Auto-RP - C-RP announcement filter
+----------------------------------------------------------
|     access-list 44 permit 224.0.0.0 7.255.255.255
|     ip pim rp-address 1.1.1.1 44                         - Configures the RP statically
|                                                          - [44] Statically specify 1.1.1.1 to be RP for ACL-44 groups
|
```

- Auto-RP filtering
 > C-RPs can limit their RP announcements to include ONLY certain multicast groups.

CONFIG-SET: Auto-RP c-RP announcement filter

```
+-----------------------------------------------------
|     access-list 4 permit 224.0.0.0 7.255.255.255          - Permit all multicast traffic in 224.0.0.0/5
|     !
|     ip pim send-rp-announce Loopback0 scope 16 group-list 4 interval 10
|                                                           - This c-RP will ONLY announce being RP for the ACL-4 groups
|
```

 > A MA can filter the c-RPs and their c-RP advertisements.
!!NOTE!! The MA's "rp-announce-filter" MUST match announcements specified by the c-RP with "send-rp-announce".
!!NOTE!! When not matching the errors can be seen on the MA by using a "debug ip pim".

CONFIG-SET: Auto-RP - MA filtering c-RPs

```
+----------------------------------------------
|     ip access-list standard R2-LOOPBACK                   - ACL specifies the RP loopback
|      permit 192.1.2.2
|     ip access-list standard R2-GROUPS                     - ACL specifies the RPs groups
|      permit 224.0.0.0 7.255.255.255
|     !
|     ip access-list standard OTHER-GROUPS                  - ACL denying all other groups
|      deny   224.0.0.0 15.255.255.255
|     !
|     ip access-list standard OTHER-RPs                     - ACL specifies all other RPs
|      deny   192.1.2.2
|      deny   192.1.4.4
|      permit any
|     !
|     ip pim rp-announce-filter rp-list R2-LOOPBACK group-list R2-GROUPS
|                                                           - Accept 224.0.0.0/5 from R2
|     !
|     ip pim rp-announce-filter rp-list OTHER-RPs group-list OTHER-GROUPS
|                                                           - Deny all other groups from all other RPs
|
```

CONFIG-SET: Two-ways to Filter Auto-RP Messages with the Multicast BoundaryCommand

```
+----------------------------------------------------------------------------------
|     access-list 1 deny 224.0.1.39
|     access-list 1 deny 224.0.1.40
|     access-list 1 permit 224.0.0.0 15.255.255.255
|     !
|     interface fa0/0
|     ip multicast boundary 1                               - Older IOS'srequire an ACL
|     ip multicast boundary filter auto-rp                  - Newer IOS's don't require an ACL
|
CONFIG-SET: Filter Admin Multicast Groups while Allowing IGMP Joins to be received
+-------------------------------------------------------------------------------------------
|      access-list 1 permit 239.0.0.0 0.255.255.255
|      !
|      interface fa0/0
|       ip multicast boundary 1                              - Filters all admin-scoped multicast traffic beyond e0/0
|

- BSR-Border Filtering
 > Allows exchange of PIM message, but prevents BSR messages from being sent or received through an interface.
 > Configured on the interfaces with "ip pim bsr-border".

- PIM Neighbor Filtering
 > Restricts PIM neighbor establishments on an interface, while still allowing multicast clients to join groups.
 > Configured with "ip pim neighbor-filter {acl}".

- Group-to-RP Filtering
 > Used to limit the client join/prune messages destined for the specified RP and for a specific list of groups.
 > Usually configured on clients such that they ignore the RP they don't trust.
 > Configured with "ip pim accept-rp {rp-address | auto-rp} [access-list]".

- IGMP Join Filtering
 > By default a host can join any multicast group it wishes to on a segment running IP multicast routing.
 > To control which groups hosts may join, configure an ACL with the command "ip igmp access-group".

- Multicast Route-Limiting
 > Used to limit the number of mroutes that may be added to the multicast routing table.
 > Configured with the command "ip multicast route-limit".

- Multicast Rate-Limiting
 > Controls the sending rate from the source to a multicast group.
 > Configured with "ip multicast rate-limit".

-----------
 COMMANDS
-----------
#ip pim rp-address {ip} [acl] [override]        - Statically configures the RP on all routers including the RP
                                                - [ACL] Limit the groups a RP will advertise
                                                - [override] Overrides dynamically learnt RP mappings

#ip pim send-rp-announce {src-int} scope {ttl}  [group-list {acl} interval {sec}]
                                                - Defines each c-RP
                                                - {int} The IP address to advertise as the candidate RP
                                                - {ttl} Is the scope TTL of the advertisement message
                                                - {acl} Announce being RP for limited groups
                                                - {interval} How often the candidate announcements are sent

#ip pim rp-announce-filter rp-list {rp-acl} group-list {group-acl}
                                                - Enables the MA to only accept certain groups from certain c-RP
                                                - {rp-acl} ACL listing the RP/s allowed/denied
```

```
                                                    - {group-acl} ACL listing the multicast group allowed/denied

#ip multicast boundary {acl}                        - Filters all multicast traffic matching the ACL
#ip multicast boundary filter auto-rp               - Filters only auto-RP announce and discovery messages

#ip pim accept-rp {rp-address | auto-rp} [acl]      - Limits client join/prune messages for specific RPs
                                                    - Configured on clients to ignore the RP they don't trust

#ip multicast route-limit {amount}                  - Limits the number of mroutes that is allowed to be added to the multicast table

#interface fa0/0
 #ip pim bsr-border                                 - Allows exchange of PIM message, but not BSR messages
 #ip pim neighbor-filter {acl}                      - Restricts PIM neighbor establishments on the interface
 #ip igmp access-group {acl}                        - Restricts the multicast groups that hosts may join on an interface
 #ip multicast rate-limit {in | out} [group-acl] [source-acl] {kbps}
                                                    - Controls the sending rate from the source to a multicast group
                                                    - {in} Accepts a rate of the kbps value or slower on the interface
                                                    - {out} Sends only a maximum of the kbps value on the interface
                                                    - {group-acl} Which multicast groups are subject to the rate limit
                                                    - {source-acl} Controls which senders are subject to the rate limit
                                                    - {kbps} Transmission rate. (Default = 0)
                                                          Any packets greater than this value are silently discarded


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
  Multicast Scoping
*======================*
- TTL Scoping
 > With TTL scoping, routers compare the TTL value on a multicast packet with the configured TTL value of each outgoing interface.
 > If the packet TTL >= the interface TTL, the packet is forwarded.
 > If the packet TTL < the interface TTL, the packet is dropped.
 > TTL scoping is limited because the configured interface TTL applies to all multicast packets.

- Administrative scoping
 > Refers simply to using the "ip igmp access-group" to filter.
 > Typically used to provide a border to keep internal multicast traffic from drifting out of the intranet.

-----------
 COMMANDS
-----------
#access-list 1 deny   239.0.0.0 0.255.255.255
#access-list 1 permit any
#interface e0/0
 #ip igmp access-group 1                            - Denies any multicast traffic in the administrative scope

 #ip multicast ttl-threshold {value}                - Any packet's TTL lower than the specified threshold are not forwarded
```

Additional Multicast Features

- Multicast Helper
 > DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring an Intermediate IP Multicast Helper between Broadcast-Only Networks

 > When a multicast-capable network is between two subnets with broadcast-only capable hosts, the broadcast traffic could be converted to multicast traffic at the first hop router and converted back to broadcast at the last hop router to deliver the packets to the destination broadcast clients.
 > The multicast capability of the intermediate multicast network could be used for transport
 > This feature prevents unnecessary replication at the intermediate routers and can take advantage of multicast fast switching in the multicast internetwork.

```
CONFIG-SET: Multicast Helper - A broadcasts only application uses UDP-3001 between different networks
+-------------------------------------------------------------------------------------------------------
|R1#                                                      >>> BROADCAST to MULTICAST client config <<<
|     access-list 123 permit udp any any eq 3001          - Matches the broadcast application traffic
|     !
|     ip forward-protocol udp 3001                        - Changes UDP-3001 to be processed-switched traffic
|     !
|     interface fa0/0                                     - Ingress interface receiving the broadcast traffic
|      ip multicast helper-map broadcast 239.1.1.1 123    - Converts the broadcast traffic to multicast using 239.1.1.1
|
|
|R2#                                                      >>> MULTICAST to BROADCAST client config <<<
|     access-list 123 permit udp any any eq 3001          - Matches the broadcast application traffic
|     !
|     ip forward-protocol udp 3001                        - Changes UDP 3001 traffic to be processed-switched traffic,
|     !                                                       which is required by the helper-map command
|     interface s0/0                                      - Interface receiving the multicast traffic
|      ip multicast helper-map 239.1.1.1 10.5.1.255 123   - Converts traffic back to broadcast, destination is 10.5.1.255
|     !
|     interface fa2/1                                     - Egress interface of destination broadcast traffic
|      ip directed broadcast                              - Destination interface must support directed broadcast transmission
|
```

- SDR Listener Support
 > The MBONE is the small subset of Internet routers and hosts which are interconnected and capable of forwarding IP multicast traffic.
 > Other multimedia content is often broadcast over the MBONE. Before one can join a multimedia session, one must know which multicast group address and port is being used for the session, when the session is going to be active and what sort of applications are required on one's workstation. The MBONE Session Directory Version 2 (SDR) tool provides this information.
 > By default, the switch does not listen to session directory advertisements.

```
- Load-Splitting Multicast Traffic
 > DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Load Splitting IP Multicast Traffic over ECMP
 > Describes how to load split/share IP multicast traffic over ECMP (Equal Cost Multipaths).
 > Multicast traffic from different sources or from different sources and groups are load split across equal-cost paths to take
      advantage of multiple paths through the network.

- Multicast Heartbeat
 > DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Monitoring and Maintaining IP Multicast
         > Monitoring IP Multicast Delivery Using IP Multicast Heartbeat

 > Provides a way to monitor the status of IP multicast delivery and be informed when the delivery fails via SNMP traps.


-----------
 COMMAND
-----------
#ip multicast helper-map broadcast {m-ip} {acl}       - Configures a first hop router to convert broadcast traffic to
                                                            multicast traffic
#ip multicast helper-map {group-ip} {direct-broadcast} {acl}
                                                      - Configures a last hop router to convert multicast traffic to
                                                            broadcast traffic
#ip directed-broadcast                                - Configures directed broadcasts. Required for mhelper
#ip forward-protocol udp [port]                       - Configures IP to forward the used protocol. Required for mhelper

#ip sdr listen                                        - Enables SDR listener support

#ip multicast multipath                               - Enables ECMP multicast load splitting based on source address

#snmp-server enable traps ipmulticast                 - Enables the router to send IP multicast traps
#ip multicast heartbeat {mgroup} {min} {window-size} {interval}
                                                      - Enables the monitoring of the IP multicast packet delivery


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
  SSM (Source Specific Multicast)
*====================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring Source Specific Multicast
```

- SSM is an extension of IP multicast where datagram traffic is forwarded to receivers from only those multicast sources to which the receivers have explicitly joined.
- For multicast groups configured for SSM, only source-specific multicast distribution trees (no shared trees) are created.

- IANA has reserved the address range 232.0.0.0/8 for SSM applications and protocols.

```
-----------
 COMMANDS
-----------
# sh ip igmp groups detail                      - Shows the (S,G) channel subscription through IGMPv3
# sh ip mroute                                  - Shows whether a multicast group supports SSM service or whether a
                                                      source-specific host report was received

#ip pim ssm [default | range-acl]               - Enables SSM by defining the SSM range of IP multicast addresses
#interface fa0/0
 #ip igmp version 3                             - Enables IGMPv3 on this interface. IGMPv3 required for SSM (def = v2)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*================================================*
  MSDP (Multicast Source Distribution Protocol)
*================================================*
```

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
    > Using MSDP to Interconnect Multiple PIM-SM Domains

- MSDP is used to interconnect multiple PIM-SM domains:
  > Allows a RP to dynamically discover active sources outside of its domain.
  > Introduces a more manageable approach for building multicast distribution trees between multiple domains.

- MSDP depends on BGP or MP-BGP (MultiProtocol BGP) for interdomain operation.
- It is recommended that MSDP is run on the RPs sending to global multicast groups.

```
-----------
 COMMANDS
-----------
# sh ip msdp summary                            - Shows the configured peers and their counters
# sh ip msdp peer                               - Shows all info regarding peer(s)

# debug ip msdp peer                            - Debugs MSDP activity for the peer-address
# debug ip msdp routes                          - Provides more detailed debugging information
# debug ip msdp detail                          - Shows the contents of source-active messages

#ip msdp peer {ip} connect{int} remote-as {asn} - Configures MSDP peer in different AS
#ip msdp peer {ip} connect {int}                - Configures MSDP peer within the same AS
```

PGM (Pragmatic General Multicast)

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
    > Configuring PGM Host and Router Assist

- PGM is a reliable multicast transport protocol for applications that require ordered, duplicate-free, multicast data delivery from multiple sources to multiple receivers.
- PGM guarantees that a receiver in a multicast group either receives all data packets from transmissions and retransmissions, or can detect unrecoverable data packet loss. PGM is intended as a solution for multicast applications with basic reliability requirements. It is network-layer independent.
- The Cisco implementation of PGM Router Assist supports PGM over IP.
- This feature uses a TSI (Transport Session Identifier) that identifies a particular PGM session.
- The PGM Router Assist feature saves substantial bandwidth.

COMMANDS

```
# sh ip pgm router                                  - Shows information about PGM traffic statistics and TSI state.
# clear ip pgm router                               - Clears the PGM traffic statistics.

#interace fa0/0
 #ip pgm router                                     - Enables the router to assist PGM on this interface
```

MRM (Multicast Routing Monitor)

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
    > Using the Multicast Routing Monitor

- MRM facilitates automated fault detection in a large multicast routing infrastructure.
- MRM is designed to alert a network administrator of multicast routing problems in close to real-time.
- MRM has two components: MRM tester and MRM manager. MRM tester can be a sender or a receiver.
- Only the MRM testers and managers need to be running the MRM-supported Cisco IOS version.

```
                               |            |--- (e0) RECEIVER
       SENDER (e0) ------|            |
                               |-----------|
                                     |
                                     |(e0)
                               TEST MANAGER
```

```
CONFIG-SET: MRM (Multicast Routing Monitor)
+-----------------------------------------------------
|     interface Ethernet0
|      ip mrm test-sender                           - Test sender configuration
|
+-----------------------------------------------------
|     interface Ethernet0
|      ip mrm test-receiver                         - Test receiver configuration
|
+-----------------------------------------------------
|     access-list 1 permit 10.5.1.2                 - Matches the sender's address
|     access-list 2 permit 10.5.4.2                 - Matches the receiver's address
|     !
|     ip mrm manager test1                          - Test manager configuration
|      manager e0 group 239.1.1.1
|      senders 1
|      receivers 2 sender-list 1
|
+-----------------------------------------------------
| The MRM manager is not started by default. Start the manager with "mrm start".
>
>     Test_Manager# show ip mrm manager
>        Manager:test1/10.5.2.2 is not running
>         Beacon interval/holdtime/ttl:60/86400/32
>         Group:239.1.1.1, UDP port test-packet/status-report:16384/65535
>          Test sender:
>            10.5.1.2
>          Test receiver:
>            10.5.4.2
>
>     Test_Manager# mrm start test1
>      *Feb  4 10:29:51.798: IP MRM test test1 starts ......
>
| The test manager sends control messages to the test sender and the test receiver as configured in the test parameters.
| The test receiver joins the group and monitors test packets sent from the test sender.
|
>     Test_Manager# show ip mrm status
>     IP MRM status report cache:
>     Timestamp        Manager          Test Receiver   Pkt Loss/Dup (%)      Ehsr
>     *Feb  4 14:12:46 10.5.2.2         10.5.4.2        1            (4%)     29
>     *Feb  4 18:29:54 10.5.2.2         10.5.4.2        1            (4%)     15
>
```

```
-----------
 COMMANDS
-----------
# sh ip mrm status                              - Shows MRM status and counters
# sh ip mrm manager                             - Shows manager group, status, test sender and receiver
# mrm start {name}                              - Starts the MRM manager


#ip mrm manager {name}                          - Creates/edits an MRM manager
 #manager {src-int} group {ip}                  - Specifies the managers source group IP address
 #senders {acl}                                 - Configures test sender request parameters
 #receiver {acl} sender-list {acl}              - Configures test receiver request parameters and test senders to be monitored

#interface fa0/0
 #ip mrm {test-sender|test-receiver}            - Configures a sender or a receiver
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================================*
```

MVR (Multicast VLAN Registration)

```
*====================================================*
```

- DOC-CD LOCATION
  > Switches - LAN
  > Cisco Catalyst 3560 Series Switches
  > Configuration Guides
  > Configuring IGMP Snooping and MVR
  > Configuring MVR

- MVR allows the single multicast VLAN to be shared in the network while subscribers remain in separate VLANs.
- MVR provides the ability to continuously send multicast streams in the multicast VLAN, but to isolate the streams from the subscriber VLANs for bandwidth and security reasons.
- In multicast VLAN networks, subscribers to a multicast group can exist in more than one VLAN.
- If the VLAN boundary restrictions in a network consist of layer2 switches, it might be necessary to replicate the multicast stream to the same group in different subnets, even if they are on the same physical network.
- MVR routes packets received in a multicast source VLAN to one or more of the receiver VLANs.
- Clients are in the receiver VLANs and the multicast server is in the source VLAN.
- Copy descretly owned by Kane Bagwell

- Guidelines and Limitations:
  > Receiver ports can only be access ports; they cannot be trunk ports.
  > Receiver ports on a switch can be in different VLANs but should not belong to the multicast VLAN.
  > Only one MVR multicast VLAN per switch is supported.
  > Do not configure MVR on private VLAN ports.
  > MVR data received on an MVR receiver port is not forwarded to MVR source ports.
  > All source ports on a switch belong to the single multicast VLAN.

```
-----------
 COMMANDS
-----------
```

```
# sh mvr                                      - Shows the MVR status and values for the switch
# sh mvr interfaces                           - Verifies the flow of the multicast stream
# sh mvr member                               - Lists who subscribes to the multicast group

#no ip multicast-routing distributed           - Disables multicast routing globally on the switch
#mvr                                          - Enables MVR globally
#mvr group {MGROUP-IP} {count}                - Specifies the multicast group where the stream is sent
#mvr vlan {vlad-id}                           -(o) Specifies the VLAN in which multicast data is received
                                              - All source ports must belong to this VLAN, (def VLAN=1)
#int gi0/1
 #mvr type source                             - Configures the port receiving multicast data as source ports

#int range fa0/15-20
 #mvr type receiver                           - Configures the ports where subscribers are connected to
 #mvr vlan {vlan-id} group {MGROUP-IP}        -(o) Statically configure a port to receive the multicast IP address traffic
 #mvr immediate                               -(o) Enable the immediate-leave feature of MVR on the port
```

-=-

===

DVMRP (Distance Vector Multicast Routing Protocol)

===

- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
        > Configuring DVMRP Interoperability

- Cisco IOS does not support a full implementation of DVMRP; however, it does support connectivity to a DVMRP network.
- Cisco routers know enough about DVMRP to successfully forward multicast packets to and receive packets from a DVMRP neighbor.
- It is also possible to propagate DVMRP routes into and through a PIM cloud.
- The Cisco IOS software propagates DVMRP routes and builds a separate database for these routes on each router, but PIM uses this routing information to make the packet-forwarding decision.

- The major differences between PIM-DM and DVMRP are defined as follows:
 > DVMRP uses its own distance vector routing protocol that is similar to RIPv2. It sends route updates every 60 sec and considers 32 hops to be infinity. Use of its own routing protocol adds more overhead to DVMRP operation compared to PIM-DM.
 > DVMRP uses probe messages to find neighbors using the All DVMRP Routers group address 224.0.0.4.
 > DVMRP uses a truncated broadcast tree, which is similar to an SPT with some links pruned

----

COMMANDS

```
#interface fa0/0
 #ip dvmrp metric {metric} [list {acl}] [protocol] [route-map]
                                              - Configures the metric associated with a set of destinations for DVMRP reports
                                              - [route-map] Subjects unicast routes to route-map conditions before
                                                     they are injected into DVMRP.
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
  Troubleshooting Multicast
*==============================*

- For general troubleshooting, consider the following:
 > Have you tried emulating an IGMP join on a client's router interface?                   # sh ip igmp groups
 > Have you tried emulating the multicast traffic from the source?                         # ping {m-ip}
 > Do all the transit routers have multicast-routing enabled?                              # sh ip multicast | i Routing
 > Are you sending and receiving multicast traffic on an interface?                        # sh ip pim int {int} stats
 > Does the router connected to the source list the join membership reports?               # sh ip igmp group {int}
 > Is the same IGMP version used throughout?                                               # sh ip igmp interface
 > Do all the transit interfaces have the correct PIM-mode enabled?                        # sh ip pim int
 > Are the expected PIM neighbors showing?                                                 # sh ip pim neighbor
      >> If not, are any stub filters configured (look at 'PIM neighbor filter')?          # sh ip pim int {int} detail
 > Are the expected mroute entries showing?                                                # sh ip mroute {m-ip}
 > Are there any issues with the multicast fast-switching cache entries?                   # sh ip mcache {m-ip}
 > Is the expected multicast path taken from a source to a group?                          # mtrace {src-ip} {m-ip}
      >> If you want more information about the path, use this command                     # mstat {src-ip} {m-ip}
 > Are any interface TTLs exceeded on transit routers (look at 'bad hop count')?           # sh ip traffic
 > Is there a limit on the number of allowed multicast routes?                             # sh ip multicast | i limit
 > Are there any input packet drops for multicast flows?                                   # sh int {int} | i flushes
      >> If so, increase the SPT value to infinity.                                        #ip pim spt-threshold infinity


- When troubleshooting sparse mode, consider the following:
 > Was a static RP configured correctly on all routers?                                    # sh ip pim rp mapping
 > Should a static RP be preferred over a dynamically learned RP?                          # sh run | i rp-add.*override
 > Is the dynamically-chosen RP the expected RP?                                           # sh ip pim rp mapping
 > If auto-RP is used:
      >> Was sparse-dense mode enabled on the interfaces?                                  # sh ip pim int
      >> Or was auto-RP listener configured?                                               # sh run | i line|listener
 > Confirm RP reachability to all the multicast routers.                                   # ping {rp-ip}
 > Does the RP know about the source traffic (S,G)?                                        # sh ip mroute
 > Does the RP and transit routers list the clients/destinations (*,G)?                    # sh ip mroute
 > Does the elected DR know the RP's IP-address?                                           # sh ip pim rp mapping
      >> Confirm the elected DR is correctly placed and forwarding the PIM register traffic to the RP.


- When troubleshooting RPF failures, consider the following:                               # sh ip rpf
 > Has the 'RPF failed' counter increased on any router?                                   # sh ip mroute count
 > Are the expected incoming interface and outgoing interfaces listed?                     # sh ip mroute
 > Is the incoming multicast interface the next-hop back to the unicast source?            # sh ip route {src-ip}
 > Confirm the unicast source interface was enabled for multicast.                         # sh ip pim int {int}
 > For multiple paths, was RPF check enabled across equal-cost paths?                      # sh ip multicast | i Multi
 > As a last resort use a debug to find the cause.                                         # debug ip mpacket {m-ip}
      >> Remember in order to see transit traffic, disable multicast route-cache!          #no ip mroute-cache
```

```
- Consider the following solutions to RFP failures:
 > Change the unicast routing to match the expected incoming interfaces.
 > Use a static multicast route to receiving multicast traffic on a specific interface.
 > In some scenarios influencing the tree type could be used as a workaround.


- Is there a NON-broadcast or unicast only network between the source and a group?
 > If so, configure PIM over a GRE tunnel.
 > The tunnel source and destination should NOT be routed via the tunnel.


- Logging Error:
 > %PIM-6-INVALID_RP_JOIN : Received (*,224.1.1.1)
 > Could be caused by
      >> Wrongly configured static RP mappings                                     # sh run | i rp
      >> Client is filtering the accepted RPs                                      # sh run | i pim.*accept
 _   ______           ___
| | |  ___ \         / __)
| | | |___) )  _   _| (__
| | |  ____/  | | | |  _ \
| | | |        \ v /| |_| )
|_| |_|         \_/  \___/
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-*
|       INDEX             |
*-=-=-=-=-=-=-=-=-=-=-=-=-*
```

```
- Overview
      + Advantages of IPv6 over IPv4
      + PMTU (Path Maximum Transmission Unit)
- Addressing
      + Global Unicast
      + Link Local
      + Site Local
      + Unique Local
      + EUI-64
      + IPv6 Multicast
      + Anycast
      + IPv4-Compatible IPv6
      + Address Conversion for 6-to-4 tunnels
- ICMPv6
      + Router and Neighbor Discovery
      + IPv6 ICMP Rate Limiting
- IPv6 on 3560
- IPv6 over Frame Relay
- IPv6 Routing Overview
- RIPng
- EIGRP for IPv6
- OSPFv3
      + OSPFv3 authentication
      + New LSAs
- MP-BGP
      + Capabilities
      + AFI (Address Families Identifier)
      + SAFI (Subsequent Address Families Identifier)
- Tunneling & Transitioning Techniques
      + IPv6IP
      + GRE
      + Automatic 6to4
      + ISATAP
      + NAT-PT (Protocol Translation)
- IPv6 Multicast
      + Well-known Addresses
      + Ethernet Mapping
      + MLD (Multicast Listener Discovery)
      + IPv6-PIM
      + BSR
```

```
- Access-List Filtering
- Static IPv6 DNS Entries
- Troubleshooting IPv6



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
     Overview
*======================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
         > Cisco IOS IPv6 Configuration Guide, Release 12.4T

- Advantages of IPv6 over IPv4:
 > Larger address space, IPv6 has 128 bits compared to the 32 bits in IPv4.
 > Address scopes are new to IPv6.
 > Stateless address auto-configuration.
 > Multicast is part of the base specifications in IPv6, unlike IPv4.
 > No more broadcasts.
 > Faster and simpler forwarding.
 > No IPv6 header checksum.
 > Simplified header: IPv4 header (12 fields) vs IPv6 header (5 fields).
 > New flow label field in header.
 > Fixed packet header sizes, 40-bytes IPv6 compared to 20-bytes+ for IPv4.
 > Fragmentation mandatory on clients with PMTU.
 > Mobile IPv6 allows a mobile node to change its locations and addresses seamlessly.
 > Network-layer security through native IPSEC.

- PMTU (Path Maximum Transmission Unit)
 > Enabled by default for IPv6.
 > With IPv6 fragmentation is mandatory on clients through PMTU.
 > Fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to
       accommodate the size of the packets.



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
     Addressing
*======================*
- IPv4: x.x.x.x
 > Each octet(x) denotes 1 byte.

- IPv6: xxxx:xxxx:xxxx:xxxx : xxxx:xxxx:xxxx:xxxx
 > Each hex caracter(x) denotes a tuple(4 bits). Two tuples (2 hex characters) denotes 1 byte (8 bits).
 > 1st 8 bytes = network address portion.
 > 2nd 8 bytes = hosts addresses portion.
```

```
- Well-Known IPv6 addresses
 > ::A.B.C.D              - IPv4-compatible IPv6 address.
 > ::1                    - Loopback (127.0.0.1).
 > ::                     - Unspecified address (0.0.0.0) used for initial automatic address assignment.
 > ::/0                   - Default route.

- Global Unicast Addresses
 > 2000 - 3FFF            - Format prefix.
 > Structure consists of
      >> 48-bit Global Prefix assigned to regional registries.
      >> 16-bit Subnet ID or SLA (Site-Level Aggregator).
      >> 64-bit Host ID.

- Link-Local Addresses
 > FE80::/10              - Format prefix.
 > Nodes on a local link can use link-local addresses to communicate. They do not need globally unique addresses to communicate.
 > IPv6 routers should not forward packets that have link-local source or destination addresses to other links.


- Site-Local Addresses
 > FEC0::/10              - Format prefix.
 > RFC 3879 deprecated use of site-local addresses and replaced them with unique local address.


- Unique Local Addresses
 > FC00::/7               - Format prefix.
 > Is an IPv6 unicast address that is globally unique BUT is intended for local site communications replacing Site-Local Addresses.
 > Are not expected to be routable on the global internet but should be routable within a site/domain.
 > Structure consists of
      >> 41-bit Global identifier used to create a globally unique prefix.
      >> 16-bit Subnet identifier of a subnet within a site.


- EUI-64
 > IPv6 host addresses are generated from interface MAC addresses.
 > A MAC address is 48-bits and IPv6 host address is 64-bits.
 > The extra 16-bits are derived as follows:
      >> MAC address 1234.5678.9012
      >> Invert the 7th most significant bit (in binary) = 00010010 > 00010000 (thus 12 becomes 10)
           = 1034.5678.9012
      >> Insert FFFE in the middle
           = 1034.56FF.FE78.9012

- Multicast Addresses
 > FF00::/8               - Format prefix.
 > FF3x::/96              - SSM address range.

 > All multicast addresses begin with the format prefix 1111 1111, written as FF.
 > The format prefix, FF, is followed by two fields: flags and scope. These two fields are 4-bits each.
 > The remaining 112 bits are the group ID.
 > Well-known multicast addresses:
      >> FF02::1         - All multicast nodes on a subnet.
      >> FF02::2         - All multicast routers on a subnet.
      >> FF02::5         - OSPFv3 routers.
      >> FF02::6         - OSPFv3 designated routers.
      >> FF02::9         - RIPnG routers.
      >> FF02::A         - EIGRP routers.
      >> FF02::D         - PIM routers.

- Anycast Addresses
 > One single address assigned to a set of interfaces that belong to different nodes.
 > Using the routing table, a packet sent to an anycast address will be delivered to the closest device with that address.
 > There is no specially allocated range for anycast, as anycast addresses are allocated from the unicast address space.
 > Assigning a unicast address to more than one interface makes a unicast address an anycast address.
 > Anycast addresses must not be used as the source address of an IPv6 packet.
 > Configured with the 'anycast' keyword.

- IPv4-Compatible IPv6 Address
 > An IPv6 unicast address with all zeros in the high-order 96 bits and an IPv4 address in the low-order 32 bits of the address.
 > ::A.B.C.D               - IPv4-Compatible IPv6 address.

- IPv4 to IPv6 Conversion (needed with IPv6 6-to-4 tunnels)
 > Let's take 192.168.99.1
      1- Divide each octet by 16 (since HEX is a Base-16)
         IE 192/16 = 12 times exactly with 0 left over
         And 12 in HEX is represented as C
         Thus 192 in HEX is C0.

      2- 168/16 = 10 times with 8 left over
         And 10 in HEX is A
         Thus 168 in HEX is A8.

      3- 99/16 = 6 times with 3 left over
         Thus 99 in HEX is 63.

      4- 1/16 = 0 times with 1 left over
         Thus 1 in HEX is 01.

 > So IPv4 (192.168.99.1) = IPv6 portion to be used(C0A8.6301) which makes a full 6-to-4 address 2002:c0a8:6301:1::1/64.

- IPv6 to IPv4 Conversion
 > Let's take the IPv6 address portion of C0A8.6301:
      1- Break the address into 2 tuple groupings (2 hex characters) = C0 A8 63 01.
      2- Take C0 and multiply the first character 'C' by 16 and the second character '0' by 1.
      3- Add the two decimal values together to get the IPv4 decimal equivalent of C0 as 192 ((c=12)*16) + (0*1).
      4- Same with A8, ((A=10)*16) + (8*1) = 168.
      5- Same with 63, (6*16) + (3*1) = 99.
      6- Same with 01, (0*16) + (1*1) = 1.
      7- This will give a IPv4 address of 191.168.99.1.
```

- With IPv6, multiple IPv6 addresses can be configured per interface. There are no primary and secondary address as in IPv4.
- When pinging a link-local IP, the outgoing interface must be specified, since the same address could be used on multiple interfaces.

```
-----------
 COMMANDS
-----------
# sh ipv6 int fa0/0                              - Shows all IPv6 interface parameters
# sh ipv6 neighbor                               - Equivalent to "sh ip arp"
# sh ipv6 route                                  - Equivalent to "sh ip route"
# sh ipv6 int brief                              - Equivalent to "sh ip int brief"
# sh ipv6 traffic                                - Shows statistics about IPv6 traffic

# debug ipv6 packets                             - Shows detailed messages for IPv6 packets
# debug ipv6 nd                                  - Shows messages for IPv6 ICMP neighbor discovery

# ping ipv6 {ip} [ext-int]                       - [ext-int] Must be specified if pinging a link-local address
# telnet {ipv6} /ipv6 /source-interface {int}    - Telnetting to a link-local host required to be sourced

#ipv6 unicast-routing                            - Enables IPv6 routing
#ipv6 cef                                        - Enables CEF for IPv6 (default = disabled)
#interface fa0/0
 #mac-add 1034.5678.9012                         -(o) Used the specified MAC appose to the BIA (Built In Address)
 #ipv6 enable                                    -(o) Enable link-local EUI-64 address (auto generates link-local IP)
 #ipv6 add FE80::1 link-local                    -(o) Or manually create a link-local address

#interface fa0/1
 #ipv6 add 2001::/64 eui-64                      - Manually configures the global unicast address, and enabling EUI-64
 #ipv6 add 2001:155:1:146::1/64                  - Manually configures a full IPv6 address

#interface fa0/2
 #ipv6 add 2001:oDB8:c058:6301::/128 anycast     - Configures the anycast address

#interface fa0/3
 #ipv6 add autoconfig                            - Address is then based on stateless auto-config
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================*
   ICMPv6
*=======================*
```

- ICMPv6 Neighbor and Router Discovery
 > Replaces IPv4 ARP.

- Router discovery functionality is enabled in IPv6 such that the routers send Router-Advertisements so that IPv6 nodes can automatically discover the routers on the local link.
- Neighbor discovery in IPv6 is a way for IPv6 nodes to discover the presence of other IPv6 nodes on the same link and then to keep track of them.

- NS - Neighbor Solicitation
  >> Ask for information about neighbors.
- NA - Neighbor Advertisement
  >> Advertise to other neighbors.
- RS - Router Solicitation
  >> Ask for information about local routers.
- RA - Router-Advertisement
  >> Advertise as an active router.

- The sending of RA messages is automatically enabled on ethernet and FDDI interfaces when the IPv6 unicast-routing is enabled.
- For other interface types, the sending of RA messages must be manually configured by using "no ipv6 nd ra suppress".

- IPv6 ICMP Rate Limiting
  > A feature that implements a token bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent out onto the network.

COMMANDS

```
#ipv6 neighbor 2001::1  E0/0  1234.5678.9012     - Configures a static MAC entry in the IPv6 neighbor discovery cache
#ipv6 icmp error-interval {ms} {bucketsize}      - Limits IPv6 ICMP error messages interval and bucket size.
#no ipv6 nd ra suppress                          - Enables the sending of RA messages on non ethernet interfaces (Old command)
#no ipv6 nd suppress-ra                          - Newer command of above command
```

-=-

======================

IPv6 on 3560

======================

- DOC-CD LOCATION
  > Switches > LAN-Switches
   > Cisco Catalyst 3560 Series Switches
    > Configuration Guides
     > Catalyst 3560 Switch Software Configuration Guide, Rel. 12.2(25)SEE
      > Configuring SDM Templates

- Configuration steps
  > Confirm the configured SDM (Switch Database Manager) template.
  ```
  # sh sdm prefer
  ```
  > Change the SDM template to support IPv4 and IPv6.
  ```
  #sdm prefer dual-ipv4-and-ipv6 {default | routing | vlan}
  ```
  > Then reload the switch.

COMMANDS

```
# show sdm prefer                                - Will display the current SDM profile and statistics

#sdm prefer dual-ipv4-and-ipv6 default           - Changes SDM template to support IPv6
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   IPv6 over Frame-Relay
*=========================*
- NBMA (Non Broadcast MultiAccess) networks:
 > Requires static resolution on multipoint interfaces.
 > This is required for global unicast addresses and link-local addresses otherwise recursion will break.
 > Inverse neighbor discovery (similar to InARP) is not yet implemented.

-----------
 COMMANDS
-----------
#sh frame-relay map                             - Shows the DLCI mappings, status, dynamic/static, LMI types
#sh frame-relay pvc [dlci]                      - Shows the DLCI status, messages, packets TX/RX

#ipv6 unicast-routing                           - Enables IPv6
#interface se0/0
 #ipv6 add 2001:155:1::5/64
 #frame map ipv6 2001:155:1::3 503 broadcast    - Configures static layer3-to-layer2 mapping for the global unicast address
 #frame map ipv6 FE80::3 503                    - Configures static layer3-to-layer2 mapping for the link-local address
#interface se0/1
 #ipv6 address FE80::1 link-local               - Manually create a link-local address
#interface se1/1.100 point-to-point
 #ipv6 address 2001:10:1::/64 EUI-64            - Creates a global unicast address with EUI-64
 #frame-relay interface-dlci 102                - Maps the DCLI to the interface


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
  IPv6 Routing Overview
*========================*
- IPv6 unicast routing is disabled by default.
- IPv6 static routing has the same implications as IPv4 static routing
 > If routed to a next-hop IP, the next-hop is resolved recursively to an exit interface.
 > Multipoint interfaces resolve the final destinations.
 > Point-to-point links require no next-hop resolution.

- Static to next-hop
 > With ICMP ND (Neighbor Discovery) there is no proxy ability to learn the remote neighbor (such as in InARP discovery).
 > When a static route is directed out an interface, it should be pointed to the next-hop instead of the interface.

!!NOTE!! Dynamic information recurses to the remote link-local address, not to the global unicast address!

-----------
 COMMANDS
-----------
# sh ipv6 static [detail]                       - Shows information about the IPv6 static routes

#ipv6 route {address} {prefix} {int} {NH}       - Creates a static IPv6 route
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
     RIPng
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IPv6 Configuration Guide, Release 12.4T
         > Implementing RIP for IPv6

- Differences from RIPv2
 > Tags are just locally significant arbitrary numbers/words.
 > Uses UDP port 521.
 > Multicasts are sent to FF02::9.

- Similar to RIPv1/RIPv2
 > Split-horizon is enabled by default, which needs to be disabled on multipoint NBMA links.
 > Default routing.
 > Summarization.
 > Offset-list.
 > Distribute-list.

-----------
 COMMANDS
-----------
# sh ipv6 protocols                             - Shows how if RIP is enabled, and on which interfaces
# sh ipv6 rip                                   - Shows RIP protocol statistics and counters
# sh ipv6 rip next-hops                         - Shows information about the next-hop addresses
# sh ipv6 route rip                             - Shows only the RIP routes in the table
# clear ipv6 route *                            - This refreshes the routing table from the routing database
                                                - This works differently to IPv4
# clear ipv6 rip {process}                      - This will refresh the routing database
# debug ipv6 rip                                - Shows the sent and received RIPv6 updates

#ipv6 router rip {tag}                          - Enables RIPng the {tag} is locally significant
#interface fa0/0
 #ipv6 rip {tag} enable                         - Interface level command that auto enables the global RIP process
                                                - The tag number/name is locally significant
#no ip split-horizon                            - Needs to be disabled on multipoint NBMA links
#ipv6 rip TAG summary-address {prefix}          - Configures address summarization

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    IPv6 - EIGRP
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IPv6 Configuration Guide, Release 12.4T
         > Implementing EIGRP for IPv6
```

- Uses protocol number 88.
- Uses multicast address FF02::A.
- A ping to the multicast address could be used to verify IPv6 neighbors.
- To configure EIGRP for IPv6, you must enable IPv6 on the interface and enable the EIGRP routing process.
- EIGRP for IPv6 has a shutdown feature. The routing process should be "no shut" to start.
- The router-id used for the IPv6 EIGRP process is still a 32-bit field.
- EIGRP for IPv6 transmits hello packets with the link-local address of the transmitting interface as the source address.

```
-----------
 COMMANDS
-----------
# sh ipv6 eigrp {asn} neighbors                   - Shows the neighbors discovered, holdtime, uptime, SRTT, RTO, etc
# sh ipv6 eigrp {asn} topology                    - Shows entries in the EIGRP topology table
# sh ipv6 route eigrp                             - Shows the current EIGRP routes in the IPv6 routing table

#ipv6 router eigrp {asn}                          - Enters EIGRP configuration mode
 #router-id {32-bit value}                        - Configures a router-id
 #no shutdown                                     - Starts the EIGRP routing process

#interface fa0/0
 #ipv6 address {ip}                               - Specifies an IPv6 address
 #ipv6 enable                                     - Generates an IPv6 address
 #ipv6 eigrp {asn}                                - Enables EIGRP on the interface

 #ipv6 bandwidth-percent eigrp {asn} {%}          - Configures the bandwidth percent EIGRP may use on a interface (def = 75)
 #ipv6 summary eigrp 1 2001:0DB8:0:1::/64         - Examples of an aggregate address sent from a interface

 #no ipv6 next-hop-self eigrp {asn}               - Instructs EIGRP to use the received next-hop value instead of default
 #no ipv6 split-horizon eigrp {asn}               - Disables EIGRP for IPv6 split horizon on the specified interface


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
     OSPFv3
*======================*
```

- DOC-CD LOCATION

```
      > 12.4T Configuration Guides
       > Cisco IOS IPv6 Configuration Guide, Release 12.4T
        > Implementing OSPF for IPv6
```

- Still uses protocol number 89.
- Multicast addresses used are FF02::5 (All SPF-Routers) and FF02::6 (All DR-Routers).
- OSPFv3 has per-link, instead of per-subnet protocol processing compared to OSPFv2.
- Multiple addresses are possible per interface.
- Operation is still very similar to OSPFv2.
- One requirement is that the router-id should still be a valid IPv4 address.
  > Should either have an UP/UP interface with a IPv4 address, or
  > The "router-id" command should be used.
- Network types and timers are the same as OSPFv2.

```
- OSPFv3 Authentication
 > OSPFv3 doesn't include any authentication capabilities of its own. Instead, it relies entirely on IPv6 IPSEC.
 > IPSEC authentication can be configured either per-interface or per-area.
 > AH (Authentication Header) provides authentication via either SHA1 or MD5.
 > Note that the key lengths must be exact: 40hex digits for SHA1 or 32hex digits for MD5.
 > The key string used for the SA must be the same in each direction between two OSPFv3 neighbors.
 > The first parameter to specify is the SPI (Security Policy Index).
 > The SPI functions similarly to key numbers in a key-chain, but is communicated via AH and must match at both ends of the
     adjacency. Copy descretly owned by Kane  Bagwell
 > The SPI number is arbitrary, but must be between 256 and 4,294,967,295 (32-bit).

- There are two new LSAs (Link State Advertisements):
 > Link LSA
     >> Advertises the link-local address to all routers attached to that link.
     >> Advertises the IPv6 prefixes on the link to the routers attached to that link.
     >> Advertises OSPF options.

 > Intra-Area LSA
     >> Either associates a list of IPv6 prefixes with a transit network by referencing a network LSA.
     >> Or associates a list of IPv6 prefixes with a Router-By referencing a router LSA.

- LSA flooding scopes have also changed to
 > Link-Local scope.
 > Area scope.
 > AS (Autonomous System) scope.
```

```
-----------
 COMMANDS
-----------
# show ipv6 ospf neighbors                      - Shows the OSPF neighbors
# show ipv6 ospf database                       - Shows all the LSA's for each area
# show ipv6 ospf interface                      - Shows the authentication method used
# show crypto ipsec sa                          - Shows the security associations
# show crypto ipsec policy                      - Shows an overview of the authentication policies in use

#ipv6 router ospf 1                             - Configures OSPF area authentication
 #area 0 authentication ipsec spi {spi no} {md5|sha1} {key-string}

#interface S0/0
 #ipv6 ospf {process-id} area {area-id}         - Automatically enables the global process for OSPF v3
 #ipv6 ospf neighbor {link-local}               - Manually defines a neighbor by specifying the link-local address
 #ipv6 ospf network {network type}              - Changes the OSPF interface type along with counters
 #ipv6 ospf database-filter all out             - Filters outgoing link-state advertisements (LSAs) on interface
 #ipv6 ospf authentication ipsec spi {spi no} {md5|sha1} {key-string}
                                                - Configures OPSF authentication for the interface
```

MPBGP - IPV6

- DOC-CD LOCATION
  - 12.4T Configuration Guides
    - Cisco IOS IPv6 Configuration Guide, Release 12.4T
      - Implementing Multiprotocol BGP for IPv6

- Only one BGP process is allowed per router, thus IPv6 configuration is done using the address-family configuration.
- RFC 2858 defines extensions to BGP-4 which enables it to carry multiple network layer protocols.
- The multi-protocol extensions are negotiated between BGP peers using an optional capabilities parameter in the BGP Open message.

- Multi-protocol extensions for BGP-4 defines two new BGP optional transitive attributes used to advertise or withdraw routes.
- The attributes are MP_REACH_NLRI and MP_UNREACH_NLRI (Network Layer Reachability Information).
- The first two fields in these new attributes contain the AFI and the SAFI values.
- The AFI (Address Family Identifier) value identifies the network layer protocol.
- The SAFI (Subsequent Address Family Identifier) value identifies additional information about the type of NLRI carried.
- When the BGP peers exchange the multiprotocol extension capability, they also exchange AFI and SAFI numbers to identify what the other BGP peer is capable of.

- AFI Values:

> AFI 1	- IPv4
> AFI 2	- IPv6

- SAFI Values:

> SAFI 1	- Unicast
> SAFI 2	- Multicast
> SAFI 3	- Unicast and Multicast
> SAFI 4	- MPLS Label
> SAFI 128	- MPLS Labeled VPN.

- E.g. If BGP is carrying IPv6 traffic, AFI equals 2, and SAFI equals 1 for unicast, or SAFI equals 2 for multicast.

- The implementation of multiprotocol extensions within BGP are known and configured as address-families (also known as contexts):
  - "address-family ipv6" - Enters and configuration the IPv6 BGP context parameters.

- Normal BGP rules still apply for MP-BGP
  - MPBGP requires an underlying IGP for transport.
  - iBGP loop prevention:
    - iBGP-learned routes are not advertised to other iBGP neighbors.
    - Exceptions are route-reflection or confederations.
  - eBGP loop prevention:
    - Routes are not accepted if the local AS is listed in the received AS-path.
  - The same best-path selection process is used with the same BGP attributes.

- An IPv6 neighbor must be activated under the IPv6 address-family, which is disabled by default(unlike IPv4).
  - If not activated the neighbor will only exchange IPv4 routes.

```
-----------
 COMMANDS
-----------
# sh ipv6 bgp summary                           - Similar to the IPv4 command. Older command, it will be deprecated
# sh bgp ipv6 summary                           - Newer command to accomplish the same as previous command
# sh bgp ipv4 unicast summary                   - Newer IPv4 equivalent of "sh ip bgp summary"
# sh bgp ipv6 unicast                           - Shows the IPv6 BGP table
# sh bgp ipv6 unicast {prefix}                  - Shows details related to the specified prefix
# debug bgp all                                 - Shows the states, capabilities negotiation, AFT/SAFI, holdtime

#Router-Bgp {asn}
 #neighbor {ipv6 ip} remote-as 100              - Configures a neighbor using IPv6 transport
 #neighbor {ipv6 ip} update-source lo0          - Specifies source address for the session
 #address-family ipv6
  #neighbor {ipv6 ip} activate                  - Enables negotiation of IPv6 address-family for the neighbor
  #neighbor {ipv6 ip} route-reflector-client    - Enables RR for the neighbor
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================================*
   Tunneling & Transitioning Techniques
*=========================================*
```

- DOC-CD LOCATION
  - > 12.4T Configuration Guides
    - > Cisco IOS IPv6 Configuration Guide, Release 12.4T
      - > Implementing Tunneling for IPv6

- Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure.

- Manual - IPv6IP
  - > Usage: A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone.
  - > Can carry IPv6 packets only.
  - > Least overhead of all tunnel methods, but has no CLNS transport (IS-IS).
  - > Uses protocol 41.
  - > Tunnel source address should be an IPv4 address, or reference an IPv4 interface with IP-unnumbered.
  - > Tunnel destination address should be an IPv4 address.
  - > Tunnel interface address should be an IPv6 address.
  - > Configuration tunnel mode 'ipv6ip'.

CONFIG-SET: Configuring Manual IPv6-IP Tunnel on Router-A

```
+-----------------------------------------------------------
|     interface ethernet 0
|      ip address 192.168.99.1 255.255.255.0
|     !
|     interface tunnel 0
|      ipv6 address 3ffe:b00:c18:1::3/127
|      tunnel source ethernet 0                         - This should be Router-B destination address
|      tunnel destination 192.168.30.1                  - Router-B source address
|      tunnel mode ipv6ip                               - Specifies the tunnel mode
```

```
- Manual GRE/IPv4 Compatible
 > Usage: Simple point-to-point tunnels that can be used within a site, or between sites.
 > Can carry IPv6, Connectionless Network Service (CLNS), and many other types of packets.
 > Is the default tunnel mode when configuring a tunnel interface.
 > Uses protocol 47.
 > Tunnel source address, should be a IPv4 address or reference an IPv4 interface.
 > Tunnel destination address should be an IPv4 address.
 > Tunnel interface address should be an IPv6 address.
 > Configuration tunnel mode 'gre ipv6'.

CONFIG-SET: Configuring IPv6 GRE tunnel on Router-A
+------------------------------------------------------
|     interface tunnel 0
|      ipv6 address 3ffe:b00:c18:1::3/127
|      tunnel source 192.168.20.1                         - This would be Router-B destination address
|      tunnel destination 192.168.30.1                    - Router-B ethernet 0 address
|      tunnel mode gre ipv6

- Automatic 6to4
 > Usage: Allows an isolated IPv6 domain to be connected over an IPv4 network to remote IPv6 networks.
 > Unlike manual tunnels, 6to4 is point-to-multipoint.
 > Sites use addresses from the 2002::/16 prefix, where the format is 2002:border-router-IPv4-address::/48.
 > The IPv4 address, embedded in the IPv6 address, is used to find the other end of the automatic tunnel.
 > Tunnel source address should be an IPv4 address or reference an IPv4 interface.
 > Tunnel destination address is not required as this is a point-to-multipoint tunneling type. The IPv4 destination address is
       calculated, on a per-packet basis, from the IPv6 destination.
 > Tunnel interface address should be an IPv6 address. The prefix must embed the tunnel source IPv4 address.
 > Configuration tunnel mode 'ipv6ip 6to4'.

CONFIG-SET: Configuring IPv6 Automatic 6to4 Tunnel
+------------------------------------------------------
|     interface ethernet0
|      description IPv4 uplink
|      ip address 192.168.99.1 255.255.255.0
|     !
|     interface ethernet1
|      description IPv6 local network 1
|      ipv6 address 2002:c0a8:6301:1::1/64                - Subnet 1 of the IPv6 major address range
|     !
|     interface ethernet2
|      description IPv6 local network 2
|      ipv6 address 2002:c0a8:6301:2::1/64                - Subnet 2 of the IPv6 major address range
|     !
|     interface tunnel0
|      description IPv6 uplink
|      ipv6 address 2002:c0a8:6301::1/64                  - IPv4 address converted to HEX: c0.a8.63.01 (covered in beginning)
|      tunnel source Ethernet 0                           - Then into IPv6: 2002:c0a8:6301::1
|      tunnel mode ipv6ip 6to4
|     !                                                   - Ensures any other traffic to 2002::/16 is directed to tunnel
|     ipv6 route 2002::/16 tunnel 0                            interface 0 for automatic tunneling
```

```
- ISATAP
 > Usage: Point-to-multipoint tunnels that can be used to connect systems within a site.
 > Sites can use any IPv6 unicast addresses.
 > Supports automatic host-to-Router-And host-to-host tunneling.
 > ISATAP is designed for transporting IPv6 packets within a site, not between sites.
 > The ISATAP router provides standard Router-Advertisement network configuration, which allows clients to automatically
      configure themselves.
 > The interface identifier is created in modified EUI-64 format in which the first 32 bits contain the value 0000:5EFE
      to indicate that the address is an IPv6 ISATAP.
 > The IPv4 address is encoded in the last 32 bits of the IPv6 address, enabling automatic IPv6-in-IPv4 tunneling.
 > Deriving the ISATAP address:
      >> The prefix is 2001:0DB8:1234:5678::/64 and the embedded IPv4 address is 10.173.129.8 in hexadecimal as 0AAD:8108.
      >> will give the following address 2001:0DB8:1234:5678:0000:5EFE:0AAD:8108.
 > Tunnel source address should be an IPv4 address or reference an IPv4 interface.
 > Tunnel destination address is not required as this is a point-to-multipoint tunneling type. The IPv4 destination address is
      calculated, on a per-packet basis, from the IPv6 destination.
 > Tunnel interface address should be an IPv6 prefix in modified EUI-64 format. The IPv6 address is generated from the
      prefix and the tunnel source IPv4 address.
 > Configuration tunnel mode 'ipv6ip isatap'.
```

```
CONFIG-SET: Configuring IPv6 Automatic ISATAP Tunnel
+----------------------------------------------------------
|      interface ethernet 0
|       ip address 10.27.0.1 255.255.255.0
|      !
|      interface tunnel 1
|       ipv6 address 2001:0DB8::/64 eui-64
|       tunnel source ethernet 0
|       tunnel mode ipv6ip isatap
|       no ipv6 nd suppress-ra                           - Router-Adverts are enabled to allow client auto-configuration
|                                                        - This is an old-format command, to be replaced with
|                                                              "no ipv6 nd ra suppress"
```

```
- NAT-PT (Protocol Translation)
 > DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IPv6 Configuration Guide, Release 12.4T
          > Implementing Tunneling for IPv6

 > A translation mechanism, allowing IPv6-only devices to communicate directly with IPv4-only devices and vice versa using NAT.
 > Before implementing NAT-PT, you must configure IPv4 and IPv6 on the router interfaces that need to communicate between
      IPv4-only and IPv6-only networks.

 > Static NAT-PT
      >> Uses static translation rules to map one IPv6 address to one IPv4 address.
      >> IPv6 network nodes communicate with IPv4 network nodes using an IPv6 mapping of the IPv4 address configured
             on the NAT-PT router.
      >> Example- A NAT-PT device will map the source IPv6 address of 2001:0db8:bbbb:1::1 to the IPv4 address 192.168.99.2
            and vice versa.

 > Dynamic NAT-PT
      >> Allows multiple NAT-PT mappings by allocating addresses from a pool.
      >> NAT-PT is configured with a pool of IPv6 and/or IPv4 addresses.
      >> At the start of a NAT-PT session a temporary address is dynamically allocated from the pool.
      >> The number of addresses available in the address pool determines the maximum number of concurrent sessions.
      >> The NAT-PT device records each mapping between addresses in a dynamic state table.
      >> Dynamic NAT-PT translation operation requires at least one static mapping for the IPv4 DNS server.

 > Overload-PT:
      >> PAT (Port Address Translation), also known as overload, allows a single IPv4 address to be used for multiple
             sessions by multiplexing on the port number to associate several IPv6 users with a single IPv4 address.
      >> PAT can be accomplished through a specific interface or through a pool of addresses same as NAT-IPv4.

 > IPv4-Mapped NAT-PT:
      >> Sends traffic from the IPv6 network to an IPv4 network without configuring IPv6 destination address mapping.
      >> If the NAT-PT router has a NAT-PT prefix mapped, an ACL is used to find the source address for translation.
```

```
CONFIG-SET: Static NAT-PT Configuration
+----------------------------------------------
|     ipv6 unicast-routing                                - Required to be enabled
|     !
|     interface Ethernet3/1
|      ipv6 address 2001:0db8:3002::9/64                  - Interface connecting to the IPv6 only network
|      ipv6 enable
|      ipv6 nat
|     !
|     interface Ethernet3/3                               - Interface connecting to the IPv4 only network
|      ip address 192.168.30.9 255.255.255.0
|      ipv6 nat
|     !
|     ipv6 nat v4v6 source 192.168.30.1 2001:0db8:0::2     - Enables a static IPv4 to IPv6 NAT-PT mapping
|     ipv6 nat v6v4 source 2001:0db8:bbbb:1::1 192.168.30.2 - Enables a static IPv6 to IPv4 NAT-PT mapping
|     ipv6 nat prefix 2001:0db8:0::/96                    - Assigns an IPv6 prefix as a global NAT-PT prefix
```

```
-----------
 COMMANDS
-----------
# sh interface tunnel {int}                   - Shows the interfaces state, counters, etc
# sh ipv6 tunnel                              - Shows IPv6 tunnel information
# sh ipv6 nat statistics                      - Shows NAT-PT statistics
# sh ipv6 nat translations [verbose]          - Shows active NAT-PT translations
# clear ipv6 nat translation *                - Clears dynamic NAT-PT translations
# debug ipv6 nat [detail]                     - Shows debugging messages for NAT-PT translation

#interface tunnel 0                           - Configures a default mode GRE tunnel for IPV6 transport (protocol=47)
 #tunnel mode ipv6ip                          - Enables manual IPv6IP tunnel transport (protocol=41)
                                              - IPv6 is passenger and IPv4 as the encap and transport protocol
 #tunnel mode gre ipv6                        - Enables Manual IPv6 GRE tunnel transport
                                              - IPv6 is passenger, GRE the encap, IPv4 as transport protocol
 #tunnel mode ipv6ip auto-tunnel              - Enables automatic tunneling using IPv4 compatible address
 #tunnel mode ipv6ip 6to4                     - Enables automatic tunneling using 6to4
 #tunnel mode ipv6ip isatap                   - Enables automatic tunneling using ISATAP

#ipv6 nat prefix {ipv6}/{prefix}              - Assigns an IPv6 prefix as a global NAT-PT prefix
#interface fa0/0
 #ipv6 nat                                    - Enables NAT-PT on the interface

#ipv6 nat v6v4 source {ipv6} {ipv4}           - Enables a static IPv6 to IPv4 address mapping using NAT-PT
#ipv6 nat v6v4 source {list} {pool}           - Enables a dynamic IPv6 to IPv4 address mapping using NAT-PT
#ipv6 nat v6v4 pool {name} {start-ip}{end-ip}{prefix}
                                              - Specifies a pool of IPv4 addresses to be used by dynamic NAT-PT

#ipv6 nat v4v6 source {ipv4} {ipv6}           - Enables a static IPv4 to IPv6 address mapping using NAT-PT
#ipv6 nat v4v6 source {list} {pool}           - Enables a dynamic IPv4 to IPv6 address mapping using NAT-PT
#ipv6 nat v4v6 pool {name} {start-ip}{end-ip}{prefix}
                                              - Specifies a pool of IPv6 addresses to be used by dynamic NAT-PT
#interface fa0/1
 #ipv6 nat prefix {ipv6}/{prefix} v4-mapped map-acl
                                              - Allows traffic from an IPv6 network to an IPv4 network without
                                                     configuring IPv6 destination address mapping

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   IPv6 Multicast
*======================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > Cisco IOS IPv6 Configuration Guide, Release 12.4T
          > Implementing IPv6 Multicast

- All multicast addresses begin with the format prefix 1111 1111, written as FF.
- The format prefix, FF, is followed by two fields: Flags and Scope.
 > These two fields are each 4 bits.
- The remaining 112 bits are the group ID.

- Multicast address range      - FF00::/8
- SSM address range            - FF3x::/96

- Well-known Addresses:
 > FF02::1        - All multicast nodes on a subnet.
 > FF02::2        - All multicast routers on a subnet.
 > FF02::5        - OSPFv3 routers.
 > FF02::6        - OSPFv3 designated routers.
 > FF02::9        - RIPnG routers.
 > FF02::A        - EIGRP routers.
 > FF02::D        - PIM routers.
```

```
- IPv6 Multicast Mapping Over Ethernet
 > MAC address = 48-bits (6-bytes)
      >> 1st 24-bits (3-bytes)       - OUI (Organizational Unit Identifier)
      >> 2nd 24-bits (3-bytes)       - Serial number
 > The OUI for IPv4 multicast is 01:00:5E with the least significant bit of the most significant byte set.
 > The OUI for IPv6 multicast is 33:33.
 > So all IPv6 multicast addresses on ethernet will have this address format 33:33:xx:xx:xx:xx: where X is the
        last 32 bits of the 128-bit multicast address.
 > Example:
      >> Multicast address FF02::2100:FF17:FC05 will be mapped to the
                                            / /  |  \ \
         Ethernet MAC-address  33:33:FF:17:FC:05

- As with IPv4, IPv6 multicast addresses are always destinations, never source addresses.
- IPv6 nodes (hosts and routers) are required to join (receive packets destined for) the following multicast groups:
 > All-nodes multicast group FF02::1
 > All-routers multicast group FF02::2
 > Solicited-node multicast group, formed by starting with the 104-bit prefix FF02::1:FF00:0000
      and adding the lowest 24 bits of the unicast/anycast address on the end.

- MLD (Multicast Listener Discovery)
 > The IPv6 equivalent of IPv4 IGMP is called MLD, which is a sub protocol of ICMPv6.
 > MLDv2 = IGMP v3. MLDv2 therefore enables IPv6 to use the SSM operation.
 > MLD uses ICMPv6 messages in its operations.
 > MLD performs the same tasks as IGMP.
 > With MLD, routers act as queriers to determine which hosts want to receive traffic for a multicast group.
 > Hosts (including routers) are receivers that will send report messages to MLD queriers to inform them they want
        to receive multicast traffic.

- Auto-RP is not currently available. There is BSR for IPv6. As well as static configuration of an RP or embedded RP.
- IPv6-PIM (Protocol-Independent Multicast) operates the same way as IPv4-PIM with only a few differences:
 > IPv6-PIM has two modes of operation: SM (Sparse Mode) and SSM (Source-Specific Multicast).
 > IPv6 multicast does not support Dense Mode multicast.
 > There is no MSDP protocol in IPv6 multicast, since it offers alternative options such as embedded RP and SSM.
 > Requires a RP (Rendezvous Point) to be statically defined. Other routers learn about the RP through embedded info in
      MLD report messages and PIM messages.
 > SSM is derived from sparse mode and is more efficient. Uses a (S,G) model from the start to deliver multicast traffic to a group
      member from only one source which the joining host specifies, rather than all senders for that group.
 > BSR (Bootstrap Router) will automatically associate the IPv6 address of an RP with a multicast group. It will adapt to changes
      in RP mappings in case of failure.

-----------
 COMMANDS
-----------
# sh ipv6 mroute                                 - Shows the contents of the IPv6 multicast routing table
# sh ipv6 mroute active                          - Shows the active multicast streams on the router
# sh ipv6 rpf {ipv6-prefix}                      - Checks RPF information for a given unicast host address and prefix
# sh ipv6 mld groups                             - Shows the multicast groups directly connected and learned via MLD
# sh ipv6 mld interface                          - Shows multicast-related information about an interface
# sh ipv6 mld ssm-map                           - Shows SSM mapping information
# sh ipv6 pim interface                         - Shows information about interfaces configured for PIM
# sh ipv6 pim neighbor [detail]                 - Shows the PIM neighbors discovered
# sh ipv6 pim group-map                         - Shows an IPv6 multicast group mapping table
# sh ipv6 pim bsr {election | rp-cache | c-rp}  - Shows information related to PIM BSR protocol processing

# clear ipv6 pim counters                       - Resets the PIM traffic counters
# debug ipv6 mld                                - Enables debugging on MLD protocol activity
# debug ipv6 pim                                - Enables debugging on PIM protocol activity

#ipv6 multicast-routing                         - Turns multicast routing on for the router/switch
#ipv6 route {ip}/{mask} {nh} [ad] {unicast|multicast}
                                                - Configure static IPv6 uni/multicast route
#no ipv6 pim rp embedded                        - Disables embedded RP support in IPv6 PIM

                                                >>> Configuring MLD <<<
#ipv6 mld state-limit {no}                      - Limits the number of MLD states globally
#int fa1/0
 #ipv6 mld join-group {group} {incl|excl} {src} - Configures MLD reporting for a specified group and source
 #ipv6 mld static-group {group} {incl|excl}{src}- Statically forwards traffic for the multicast group as MLD joiner
 #ipv6 mld limit number {no}                    - Limits the number of MLD states on a per-interface basis
 #ipv6 mld access-group {acl}                   - Allows the user to perform IPv6 multicast receiver access control
 #ipv6 mld explicit-tracking {acl}              - Allows for the tracking of host behavior within a multicast v6 network
 #no ipv6 mld router                            - Disables MLD router-side processing on a specified interface

                                                >>> Configuring PIM <<<
#ipv6 pim rp-address {ip} [acl] [bidir]         - Statically sets the RP address
                                                - Optional for a particular group range

#ipv6 pim spt-threshold infinity                - Sets the SPT threshold to infinity to prevent switchover to the source tree
#ipv6 pim spt-threshold infinity group {acl}    - Configures when a PIM leaf router joins the SPT for the specified groups
#ipv6 pim accept-register {list | route-map}    - Accepts or rejects registers at the RP
#int fa2/0
 #no ipv6 pim                                   - Turns off IPv6 PIM on a specified interface

                                                >>> Configuring BSR <<<
#ipv6 pim bsr candidate bsr {ip}{mask} priority {no}
                                                - Configures a router to be a candidate BSR
#ipv6 pim bsr candidate rp {ip}[group][pri][scope]
                                                - Sends PIM RP advertisements to the BSR
#ipv6 pim bsr announced rp {ip}[group][pri][scope]
                                                - Announces scope-to-RP mappings directly from the BSR for the candidate RP
#int fa3/0
 #ipv6 pim bsr border                           - Configures a border for all BSMs of any scope on a specified interface
 #no ipv6 pim                                   - Turns off IPv6 PIM on a specified interface

                                                >>> Configuring SSM <<<
#ipv6 mld ssm-map enable                        - Enables the SSM mapping feature for groups in the configured SSM range
#ipv6 mld ssm-map static {acl} {Source}         - Configures static SSM mappings
#no ipv6 mld ssm-map query dns                  - Disables DNS-based SSM mapping
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
   Access-List Filtering
*============================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IPv6 Configuration Guide, Release 12.4T
         > Implementing Traffic Filters and Firewalls for IPv6 Security
```

- The standard ACL functionality in IPv6 is similar to standard ACLs in IPv4.
- The 'auth' keyword allows matching traffic against the presence of the authentication header in combination with the specified protocol TCP or UDP.

CONFIG-SET: IPv6 ACL Example

```
+----------------------------------------
|     ipv6 access-list example1
|         permit tcp any any                            - Allows any TCP traffic regardless of whether or not an AH is present
|     !
|     ipv6 access-list example2                         - Allows TCP/UDP only when AH is present(without AH no match)
|         deny tcp host 2001::1 any log sequence 5
|         permit tcp any any auth sequence 10
|         permit udp any any auth sequence 20
|     !
|     interface fastethernet0/1
|      ipv6 address 3FFE:C000:1:7::/64 eui-64
|      ipv6 enable
|      ipv6 traffic-filter example2 in                  - Applies the IPv6 ACL to the interface
|      ipv6 traffic-filter example1 out
|
```

```
 -----------
 COMMANDS
 -----------
#ipv6 access-list {name}                     - Creates the IPv6 ACL
 #{permit|deny} {prot} {IP|any|host|auth} {options}
                                             - Specifies the ACL options
#int fa0/0
 #ipv6 traffic-filer {name} {in|out}         - Applies the IPv6 ACL to the interface
#line vty 0 4
 #ipv6 access-class {name} {in|out}          - Applies the IPv6 ACL to the terminal line
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   Static IPv6 DNS Entries
*=========================*
- DNS Record Types
 > AAAA             - Maps a hostname to an IPv6 address.
 > PTR              -  Maps an IPv6 address to a hostname.


-----------
 COMMANDS
-----------
#ipv6 host {name} [port] {ipv6} {type}          - Defines a static hostname-to-address mapping
#ipv6 domain-name {name}                        - Defines the domain suffix
#ipv6 name-server {ipv6}                        - Specifies one or more hosts that supply name information
#no ipv6 domain-lookup                          - Disables DNS-based address translation (default = enabled)



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   Troubleshooting IPv6
*=========================*
- When troubleshooting IPv6, consider the following:
 > Was IPv6 enabled?                                                                        # sh run| i ipv6
 > Was IPv6 CEF enabled?                                                                    # sh ipv6 cef interface
 > Double check the typed IPv6 addresses!                                                   # sh ipv6 int brief
 > On serial interfaces, if needed, was RA (Router-Advertisements) enabled?                 # sh ipv6 int {int}| i advert
 > On the 3560 switches was the SDM template changed to support IPv6?                       # sh sdm prefer
 > For frame-relay multipoints, was a mapping configured for the link-local address?        # sh run | i frame.*FE80
 > Are any ACLs blocking protocol number 41?                                                # sh ipv6 interface | i line|list
 > IPv6, IPv6IP and GRE-IPv4 tunnels
      >> Are the tunnel source and destination IPv4 addresses?                              # sh run int tunnel {t-int}
      >> Is the tunnel address an IPv6 address?                                             # sh run int tunnel {t-int}

- When troubleshooting IGPs for IPv6, apply the same troubleshooting as with IPv4!
 > For RIPng
      >> Are the RIPng interfaces sending updates?                                          # debug ipv6 rip
      >> Was RIPng enabled on the interface?                                                # sh ipv6 rip
      >> Are RIPng routes being received and entered into the RIPng database?               # sh ipv6 rip database
      >> Do the RIPng routes appear in the table?                                           # sh ipv6 route rip
      >> Are individual routes of a summary suppressed?                                     # sh ipv6 rip | i {prefixes}
      >> If only a default route was to be sent out of an interface,
            was the 'only' keyword used?                                                    # sh run | i rip.*only

 > For EIGRP
      >> Are the interfaces correctly added to EIGRP?                                       # sh ipv6 eigrp interfaces
      >> Are the expected EIGRP adjacencies showing?                                        # sh ipv6 eigrp neighbors
      >> On multipoint interfaces, was split-horizon disabled?                              # sh run | i ipv6.*split
```

```
> For OSPFv3
     >> Are the expected adjacencies showing?                                          # sh ipv6 ospf neighbor
           >>> If not what is the cause?                                                # debug ipv6 ospf adj
     >> Is the router sending and receiving hellos?                                    # debug ipv6 ospf hello
     >> Do the timers match?                                                            # sh ipv6 ospf int {int} | i Dead
     >> Do the MTU values match?                                                        # debug ipv6 ospf adj
     >> Are any interfaces wrongly in Passive mode, due to "passive-interface default"  # sh run | i passive-int
     >> Are the interfaces configured to the correct areas?                             # sh ipv6 ospf int brief
     >> Are the network types compatible between neighbors?                             # sh ipv6 ospf int {int} | i Netw
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY

```
   _____       _____       ______
  / ___ \     / ___ \     / _____)
 | |   | |   | |   | |   ( (____
 | |   | |   | |   | |    \____ \
 | |___| |_  | |___| |    _____) )
  \_______)   \_____/    (______/
```

```
*-=-=-=-=-=-=-=-=-=-=-=-*
|      INDEX            |
*-=-=-=-=-=-=-=-=-=-=-=-*
```

```
- Overview
- QOS Packet Headers
      + IP TOS-Byte
      + DS-Field Compared
      + CS (Class-Selector)
      + EF (Expedited Forwarding)
      + AF (Assured Forwarding)
- MQC
      + Class-Maps
      + Policy-Maps
      + MQC Class-Default
      + Classification options
      + Marking options
      + Matching VOIP Traffic
      + QOS-Group
      + Nested MQC Policies
      + UPD (Unconditional Packet Discard)
- NBAR (Network-Based Application Recognition)
- Congestion Management
      + FIFO (First-In First-Out)
      + MDRR (Modified Deficit Round-Robin)
      + WFQ (Weighted Fair Queue)
      + Legacy CQ (Custom Queue)
      + Legacy PQ (Priority Queue)
      + CBWFQ (Weighted Fair Queue)
      + LLQ (Low Latency Queueing)
- Congestion Avoidance
      + Legacy WRED (Weighted Random Early Detection)
      + MQC WRED (Weighted Random Early Detection)
- Shaping
      + Legacy GTS (Generic Traffic Shaping)
      + Legacy FRTS (Frame-Relay Traffic Shaping)
      + MQC CB-Shaping (Class-Based Shaping)
      + MQC FRTS (Frame-Relay Traffic Shaping)
- Policing
      + Legacy Rate-Limit (CAR)
      + MQC Police
      + COPP (Control Plane Policing)
- RSVP
- AutoQOS
- Switching QOS
      + Classification
      + Ingress Queueing
      + Egress Queueing
      + Congestion Avoidance
      + Traffic Policing
- Compression
      + TCP Header Compression
      + STAC Compression
      + Predictor
      + RTP Header Compression
- Troubleshooting QOS
- OUTPUT 101
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
     QOS Overview
*==========================*
```

- The TX-ring/Hardware-queue is always FIFO. It can be seen with the "sh controllers | i tx_limited" command.
- QOS affects how traffic is processed in the output-queue/software-queue before the hardware-queue.
- Queueing can only be applied in an outbound direction to the interface.
- Shaping can only be applied in an outbound direction to the interface.
- Policing can be applied inbound or outbound direction to the interface.

```
-----------
 COMMANDS
-----------
# sh controllers | i tx_limit                          - Shows the TX queue length for an interface

#ip telnet tos {tos-value}                             - Changes the (default=6) telnet marking for telnets from the local router
#interface s0/0
 #tx-ring-limit {number}                               - Changes the TX-ring length for an interface
 #load-interval {sec}                                  - Sets the length of time used for load counter calculations
 #hold-queue {length} {in|out}                         - Limits the size of the IP queue on an interface
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
  QOS Packet Headers
*======================*
```

```
- IP TOS-Byte
  0   1   2   3   4   5   6   7
+-----------+---------------+---+
|           |               | C |
|  IP Prec  |     T O S     | U |
|           |               |   |
+-----------+---------------+---+
```

TOS-Byte Values = (3-bits IP PREC + 5-bits legacy)

IP Precedence Description	IP PREC Binary	IP PREC Decimal
Routine	000	0
Priority	001	1
Immediate	010	2
Flash	011	3
Flashoverride	100	4
Critical	101	5
Internetwork Control	110	6
Network Control	111	7

- DS-Field Compared

```
  0   1   2   3   4   5   6   7
+-----------|-----------+-------+
|                       |       |
|       D S C P         |  ECN  |
|                       |       |
+-----------|-----------+-------+
```

DiffServ Field Values = (6-bits DSCP + 2-bits ECN)

DSCP PHB Groups (8x + 2y)	DSCP-Field Binary (6-bits)	DSCP-Field Decimal (6-bits)	DS-Field Binary (1-byte)	DS-Field Decimal Format	DS-Field Hex Value
Default	000 000	0	000 000 00	0	0x0
CS1	001 000	8	001 000 00	32	0x20
AF11	001 010	10	001 010 00	40	0x28
AF12	001 100	12	001 100 00	48	0x30
AF13	001 110	14	001 110 00	56	0x38
CS2	010 000	16	010 000 00	64	0x40
AF21	010 010	18	010 010 00	72	0x48
AF22	010 100	20	010 100 00	80	0x50
AF23	010 110	22	010 110 00	88	0x58
CS3	011 000	24	011 000 00	96	0x60
AF31	011 010	26	011 010 00	104	0x68
AF32	011 100	28	011 100 00	112	0x70
AF33	011 110	30	011 110 00	120	0x78
CS4	100 000	32	100 000 00	128	0x80
AF41	100 010	34	100 010 00	136	0x88
AF42	100 100	36	100 100 00	144	0x90
AF43	100 110	38	100 110 00	152	0x98
CS5	101 000	40	101 000 00	160	0xA0

EF	101 110	46	101 110 00	184	0xB8
CS6	110 000	48	110 000 00	192	0xC0
CS7	111 000	56	111 000 00	224	0xE0

- CS (Class-Selector)

> Each IP precedence value gets mapped to a DiffServ value known as CS code-points.
> The CS code-points above are in the form 'xxx000'.
> The first three bits 'xxx' are the IP precedence bits for backwards compatibility, while the last 3-bits are set to zero.
> If a packet is received from a non-DiffServ aware router that used IP precedence markings, the DiffServ router can still understand the encoding as a CS code-point.

- EF (Expedited Forwarding)

> The EF traffic class is given strict priority queueing above all other traffic classes.
> The design aim of EF is to provide a low-loss, low-latency, low-jitter, end-to-end expedited service through the network.
> The EF traffic class is suitable for voice, video and other real-time services.

- AF (Assured Forwarding)

> AF behavior allows the operator to provide assurance of delivery as long as the traffic does not exceed the subscribed rate.
> Traffic that exceeds the subscription rate faces a higher probability of being dropped during times of congestion.
> The DiffServ architecture defines four separate classes in the AF PHB (Per Hop Behavior).
> Within each class (1 to 4), packets are given a drop precedence (1 to 3) (low=1, medium=2 or high=3).
> The 1st three bits of the 6-bit DSCP field define the class, the next two bits define the drop-probability, and the last bit is reserved (= zero).
> AF is presented in the format AFxy, where 'x' represents the AF-class (HIGHER class value is PREFERRED) and 'y' represents the drop-probability (HIGHER value is more likely to be DROPPED).

> AF23, for example, denotes class 2 and a high drop preference of 3.
> If AF23 was competing with AF21, AF23 will be dropped before AF21, since they in the same class and AF23 has a higher drop value.
> But if AF33 and AF21 were competing, AF33 is a more important class, therefore AF21 will be dropped first.

- A nice formula to work out the decimal value of the AF bits, is 8x+2y. Example: AF31 = (8*3) + (2*1), thus AF31 = 26.
- Alternatively, if the predefined DiffServ values are not used, any of the 64 DSCP values (0-63) can be used by configuring just that decimal value (the higher the decimal value, the higher the preference).

-=-

=====================

MQC

=====================

- DOC-CD LOCATION

> 12.4T Configuration Guides
> Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
> Part 1: Classification
> Applying QOS Features Using the MQC

- MQC is short for Modular Quality of Service CLI (Command Line Interface).
- MQC provides a framework for multiple QOS methods to be applied in the same direction on the same interface in contrast to legacy QOS mechanisms.

- Class-Maps
 > The purpose of class-maps is to classify traffic.
 > Class-map names are Case-Sensitive.
 > The match sub-commands are used to specify various criteria for classifying packets.
 > If a packet matches the specified criteria, that packet is considered a member of the class.
 > If a packet does not match the class criteria, it is evaluated against the next class.
 > Packets that fail to match any of the class-maps are classified as members of the default traffic class.
 > If more than one 'match' criterion exists in the class-map, an evaluation instruction should be specified.
 > The instruction could be one of the following: ('match-all' is the default)

```
      >> Match-any      - The traffic being evaluated by the class-map must match one of the 'match' statements.
      >> Match-all      - The traffic being evaluated by the class-map must match ALL of the 'match' statements.
```

- Policy-Maps
 > Used to configure the QOS features that should be associated with the traffic that has been classified with class-maps.
 > Policy-map names are Case-Sensitive.
 > Multiple class-maps can be referenced, which are evaluated sequentially top-down.

- MQC Class-Default
 > MQC always has a default class created named 'class-default'.
 > Any traffic not matched by a higher class will belong to the class 'class-default'.
 > If no other class-maps were defined in a policy-map, ALL traffic will belong to the class 'class-default'.

- Steps to configure MQC policies:
 1- Define traffic classifications using class-maps.
 2- Create the policy-map, and apply the QOS features to the individual class-maps.
 3- Apply the policy-map to a interface inbound or outbound.

- MQC Classification Options
 > Access-Lists
 > DSCP
 > IP Precedence
 > NBAR (see below)
 > Packet Length
 > FR-DE
 > Interface
 > QOS-Group

- MQC Marking Options
 > ATM-CLP
 > COS
 > Discard-Class
 > DSCP
 > Frame-Relay-DE

- Matching VOIP traffic can be done in two ways
 > Matching UDP/RTP headers and RTP port numbers:

```
      #class-map VOIP
       #match ip rtp 16384 16383
```

```
 > Using NBAR (specifies matching for RTP voice payload type values 0-23):
      #class-map VOIP
       #match ip rtp audio

- QOS-Group
 > An arbitrary number locally significant to the router.
 > Used when traffic passing through the router must be tagged/classified without changing anything in the packet header.

- Nested MQC Policies
 > Used to configure QOS inside other QOS policy-maps.
 > Often used on sub-interfaces as these do not have software queues associated with them.
 > To create a queue, initiate shaping in a parent policy-map, referencing the normal policy-map.
```

```
CONFIG-SET : Nested MQC Policy for the Ethernet Sub-Interface
+------------------------------------------------------------------
|     policy-map INNER-POLICY                               - Normal policy-map
|      class VOIP                                           - References the VOIP class-map
|       priority 128                                        - Reserves 128k for the VOIP class
|      class SMTP                                           - References the SMTP class-map
|       bandwidth 384                                       - Reserves 384k for the SMTP class
|     !
|     policy-map OUTER-POLICY                               - This policy will create a virtual queue to be used by QOS
|      class class-default                                  - Applies to ALL interface traffic
|       shape average cir 512000                            - Creates a queue with shaping
|       service-policy INNER-POLICY                         - References the nested policy-map
|     !
|     interface fa0/0
|      service-policy output OUTER-POLICY                   - Applies the policy to an interface
|
```

```
- UPD (Unconditional Packet Discard)
 > DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 4: Policing and Shaping
         > Modular QoS CLI (MQC) Unconditional Packet Discard
```

```
CONFIG-SET: Unconditional Packet Discard
+--------------------------------------------
|     class-map class1
|      match access-group 101                               - References ACL-101
|     !
|     policy-map policy1                                    - UPD is just a fancy name for the 'DROP' action in a policy-map
|      class class1
|       drop                                                - Any traffic matching ACL-101 will be dropped
|     !
|     interface s2/0
|      service-policy output policy1                        - Applied to the interface
|
-----------
 COMMANDS
-----------
# sh class-map [name]                          - Shows the configured class-map/s
# sh run policy-map [name]                     - Shows the configured policy-map/s
# sh policy-map interface {int}                - Shows the policy-map info and counters associated with the interface

#class-map [match-all | match-any] {name}      - Creates a class-map for classification (default = match-all)
 #match {options}                              - Specifies the various match criteria

#policy-map {name}                             - Creates a policy-map
 #class {name | class-default}                 - References previously created class-maps
  #{bandwidth | priority | shape | policy}     - Specifies a specific QOS feature for the class
  #service-policy {nested-policy}              - References nested policy-maps

#interface s0/0
 #service-policy {input | output} {policy-name} - Applies a policy-map to an interface
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================================================*
```

NBAR (Network-Based Application Recognition)

```
*=====================================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 1: Classification
         > Classifying Network Traffic Using NBAR
```

- NBAR is a classification engine that can identify traffic/protocols at an application level.
- NBAR looks into the TCP/UDP payload itself and classifies packets based on content within the payload such as transaction identifier, message type, or other similar data.
- NBAR natively supports many predefined applications/protocols, these can be seen with "match protocol ?"
- A PDLM (Packet Description Language Module) is a file that can extend the protocols that NBAR can recognize.
- New PDLMs can be downloaded from Cisco.com and can be loaded from flash memory.
- NBAR protocol discovery can be used to track and provide statistics on which protocols transit an interface.
- Custom NBAR mappings allow well-known protocols to be defined in the network as NBAR protocols with "ip nbar port-map".

```
- "match protocol http" explained:
 > Using NBAR to match HTTP traffic provides three match criterias:
      > Domain Hostname              - The URL portion between 'http://' and the first slash '/'
      > URL-entry                    - The URL portion after the first slash '/'
      > Mime type                    - The media content of a website.


 > For a list of mime-types go to: http://www.sfsu.edu/training/mimetype.htm
 > Matching website hostnames:
  #match protocol http host *facebook.com*           - This will match any hostname containing 'facebook.com'
                                                            like http://www.facebook.com or http://login.facebook.com.
  #match protocol http host *google*                 - This will match any hostname containing the word google
                                                            like http://mail.google.com or http://www.google.co.za.
  #match protocol http host google*                  - This will match http://google.co.za but not http://www.google.co.za.

 > Matching the URL string after hostname:
  #match protocol http url *.jpeg|*.jpg|*.gif        - This will match any of the URL strings with .jpeg/ .jpg/ or .gif
  #match protocol http url *.swf                     - This will match any .swf in the URL.
  #match protocol http url *video*                   - This will match http://www.cnn.com/video/index.php or
                                                             http://www.cnn.com/news/video.html.
  #match protocol http url video*                    - This will match http://www.cnn.com/video/index.php but not
                                                             http://www.cnn.com/news/video.html.


 > Matching NBAR mime categories/types:
  #match protocol http mime "image/jpg"              - This will match the JPEG mime type in the image-category.
  #match protocol http mime "image/*"                - This will match any image mime type in the image-category.
  #match prot http mime application/x-shockwave-flash - This will match all types of flash, not just .sfw.
  #match protocol http mime "application/*"          - This will match any application mime type.

-----------
 COMMANDS
-----------
# sh ip nbar port-map                          - Shows the default NBAR port mappings for applications
# sh ip nbar version                           - Shows the version of the PDLM's
# sh ip nbar protocol-discovery                - Shows traffic classes and statistics NBAR discovered

#class-map {name}
 #match protocol {protocol}                    - Matches NBAR applications in a class-map

#ip nbar pdlm {unc path}                       - Specifies where to load a new PDLM from
#ip nbar port-map custom {name} {tcp|udp} {port|range}
                                               - Maps well-known port/s of a protocol to an NBAR application
#interface s0/0
 #ip nbar protocol-discovery                   - Enables NBAR protocol discovery



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-==-*
*==========================*
  Congestion Management
*==========================*
- FIFO (First-In First-Out)
 > Is the default queueing mechanism on ethernet and serial links above 2 MB.

- MDRR (Modified Deficit Round-Robin)
 > Priority queueing mechanism for 12xx routers.
```

```
*------------------------------*
   WFQ (Weighted Fair Queue)
*------------------------------*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 2: Congestion Management
         > Weighted Fair Queueing

 > Dynamically allocates flows into queues. The allocation is not configurable, only the number of queues is configurable.
 > Guarantees throughput to all flows, and drops packets of most aggressive flows.
 > Default on Cisco interface below 2.048mb.
 > Cannot provide fixed bandwidth guarantees.
 > Configured with "fair-queue" under an interface.

-----------
 COMMANDS
-----------
# sh queueing fair                                 - Shows WFQ values

#interface s0/0
 #fair-queue [cdt] [dynamic-queues] [reserv-queues]
                                                    - Enables WFQ on an interface
                                                    - [cdt] Congestive Discard Threshold (values: 1-4096)

*------------------------------*
  Legacy CQ (Custom Queue)
*------------------------------*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 2: Congestion Management
         > Configuring Custom Queueing

 > Implementation of weighted round-robin.
 > Up to 16 configurable queues, including a priority queue.
 > Thresholds are based on the number of bytes and/or number of packets.
 > CQ is prone to inaccurate bandwidth allocations.
 > Can only apply one mechanism per interface. MQC changes this.
 > The custom queue is used to create a bandwidth reservation in the output queue based on the configured queues.
 > With the custom queue it is important to note that the behavior of the queueing mechanism only becomes evident
      when the output queue is congested.
 > Each configured queue is guaranteed only the minimum configured amount, but can utilize all unused bandwidth.
 > Because queueing is always outbound, when custom queueing is applied to the interface, no direction can be specified.
 > The queueing strategy will be 'custom-list', as seen with "sh interface".
 > Queue 0 is like a priority queue. Traffic in this queue will always be sent first.
 > 0 - 16: are configurable queues.
 > Defaults:
      >> Byte-count = 1500 bytes
      >> Queue-limit = 20 packets
```

```
-----------
 COMMANDS
-----------
# sh interface {int}                          - Shows the queueing strategy and configured queues
# sh queueing custom                          - Shows the custom queue configuration
# sh queue {int} [queue no]                   - Shows the current queue contents

#queue-list 1 protocol ip 0 udp rip           - Queue 0 is like a priority queue. Traffic in this queue will always be sent first
#queue-list 1 protocol ip 1 lt 65             - [lt] Classifies packets less than a specified size
#queue-list 1 protocol ip 1 list 177          - [list] Used to call an access list
#queue-list 1 protocol ip 2 gt 1000           - [gt] Classifies packets greater than a specified size
#queue-list 1 protocol ip 3 tcp 25            - Prioritizes TCP packets 'to' or 'from' the specified port
#queue-list 1 protocol ip 4 udp 53            - Prioritizes UDP packets 'to' or 'from' the specified port
#queue-list 1 protocol ip 5 fragments         - Prioritizes fragmented IP packets
#queue-list 1 default {queue}                 - Assigns the default queue

#queue-list 1 queue 0 limit 10                - Changes the maximum number of queue entries
#queue-list 1 queue 1 byte-count 1500         - Specifies size in bytes of a particular queue
#queue-list 1 queue 2 byte-count 640 limit 10  - Specifies both queue-limit and queue-size
#queue-list 1 queue 3 byte-count 104 limit 15
#queue-list 1 interface {int} {queue}         - Establishes priorities for packets from a named interface

#interface s0/0
 #custom-queue-list 1                         - Changes the output queueing mechanism to a custom queue

*-----------------------------*
  Legacy PQ (Priority Queue)
*-----------------------------*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 2: Congestion Management
         > Configuring Priority Queueing

 > Legacy priority queueing uses four queues (high, medium, normal and low), which get serviced from high-to-low.
 > PQ is prone to starvation.
 > The queueing strategy will be 'priority-list' as listed with "show interface" command.
 > Similarly to custom queueing, the 'gt', 'lt' and 'fragments' keywords are also available.

-----------
 COMMANDS
-----------
#priority-list 2 protocol ip high tcp telnet  - Assigns telnet traffic to the high priority queue
#priority-list 2 protocol ip medium list 100  - [list] Used to call an access-list
#priority-list 2 protocol ip normal fragments - Prioritizes fragmented IP packets
#priority-list 2 default low                  - Changes the default queue from normal to low
#interface s0/0
 #priority-group 2                            - Changes the output queueing mechanism to a priority queue
```

```
*------------------------------*
  CBWFQ (Weighted Fair Queue)
*------------------------------*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 2: Congestion Management
         > Weighted Fair Queueing

 > CBWFQ is used to reserve a guaranteed minimum bandwidth in the output queue based on each user defined class.
 > CBWFQ supports sixty four classes/queues.
 > The drop policy is tail drop or WRED, and it is configurable per class.
 > Scheduling within a single class:
      >> FIFO on sixty three classes.
      >> FIFO or WFQ on the 'class-default' class.

 > The queueing strategy only comes into effect when there is congestion in the output queue.
 > 'Class-default' needs "fair-queue" configured if "bandwidth" is not specified.
 > Weights can be defined by specifying:
      >> Bandwidth (in kbps)  - Absolute reservation based on the configured amount.
      >> Bandwidth percent    - Absolute reservation based on percentage of configured interface "bandwidth" of the link.
      >> Remaining percent    - Relative reservation based on the available interface bandwidth, not the configured "bandwidth".

 > The queueing strategy will be 'class-based queueing' as listed with "show interface" command.
 > Classification is done through ACLs or by using NBAR.
!!NOTE!! Don't forget to change the default max-reserved-bandwidth of 75% for the interface before applying the service-policy.
       "max-reserve-bandwidth" is only a configuration limitation!

-----------
 COMMANDS
-----------
# sh policy-map interface {int}                 - Shows the policy map configured with all the counters

#class-map SMTP                                 - (default = match-all)
 #match access-group SMTP                       - Uses an extended ACL to match tcp port 25
#class-map match-any HTTP
 #match protocol HTTP                           - Uses NBAR to match all http traffic
#class-map FTP                                  - Class-map names are CaSe-SeNsItIve
 #match access-group FTP

#policy-map QOS                                 - Names are CaSe-SeNsItIve, the order of the class statement are important
 #class SMTP                                    - Calls the defined class-map
  #bandwidth 512                                - Absolute reservation based on the configured amount (512k here)
 #class HTTP
  #bandwidth percent 25                         - Absolute reservation based on the % of config "bandwidth" of the link (256k here)
 #class FTP                                          since the interface has 1024k specified
  #bandwidth remaining percent 25               - Relative reservation based on what is available interface bandwidth,
                                                     not configured 'bandwidth' (1024-512-256)=256k here

#class class-default
 #fair-queue                                    - Required if "bandwidth" was not specified
```

```
#interface S0/0
 #bandwidth 1024
 #max-reserved-bandwidth {%}                    - Changes the default 75% reserved bandwidth used when queueing is applied.
 #service-policy output QOS                     - Applies queueing policy (CBWFQ) to the interface
```

```
*--------------------------------------*
  LLQ (Low Latency Queueing)
*--------------------------------------*
```

- DOC-CD LOCATION
    > 12.4T Configuration Guides
     > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
      > Part 2: Congestion Management
       > Weighted Fair Queueing

- LLQ adds the concept of a priority queue to CBWFQ, but without starving other classes.
- The LLQ provides a maximum bandwidth guarantee with low-latency, and optional burst capability.
- LLQ uses only one queue per QOS policy, but does allow multiple queues.
- LLQ has a built-in congestion-aware policer, preventing the starvation of non-priority traffic.
- The internal policer is ONLY applied during times of congestion, otherwise LLQ traffic may use any excess bandwidth.
- During times of congestion, a priority class cannot use any excess bandwidth, thus any excess traffic will be dropped.
- But during times of non-congestion, traffic exceeding the LLQ is placed into the class-default and is not priority "queued".
- This is why it is usually recommended to also add a "police" statement in the LLQ, so that priority traffic gets queued correctly or dropped.
- The queueing strategy will be 'class-based queueing' seen with the "show interface" command.

 COMMANDS

```
# sh policy-map interface {int}                 - Shows the policy-map configured with all the counters
# sh queueing int {int}                         - Shows the input and output queue size
                                                - Shows the available bandwidth that can be assigned

#class-map VOIP
 #match ip rtp 16384 16383                      - Matches RTP ports
#policy-map LLQ
 #class VOIP
  #priority {kbps} [burst {bytes}]              - Configures low-latency queueing for the VOIP class
  #police cir {bps} bc {bytes} be {bytes

 #interface S0/0
 #service-policy output LLQ                     - Applies the queueing policy to the interface
```

Congestion Avoidance

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
    > Part 3: Congestion Avoidance

- Attempts to avoid congestion before it occurs by selectively dropping traffic, i.e. random-detect.
- Weights can be based on IP precedence or DSCP values.
- WRED is typically used to avoid TCP global synchronization but is generally not too successful when the majority of flows are UDP.
- When the minimum threshold is reached WRED, becomes active and starts randomly selecting packets to be dropped.
- The rate of packet drop increases linearly as the average queue size increases until it reaches the maximum threshold.
- When the average queue size reaches the maximum threshold, the fraction of packets dropped is that of the MPD.
- When the average queue size is above the maximum threshold, all new packets are tail-dropped.
- MPD (Mark Probability Denominator)
  > Used to determine how aggressively packets will be dropped.
  > The lower the number the more aggressively dropped.
  > When the max-threshold is reached, 1/MPD will be dropped!

Legacy WRED (Weighted Random Early Detection)

- The queueing strategy will be 'random early detection (RED)', as seen with "sh interface".

COMMANDS

```
# sh queueing int {int}                        - Shows the input and output queue size and default values

#interface s0/0
 #random-detect [dscp-based | prec-based]      - Enables WRED (precedence-based is the default option)
 #random-detect prec {value} {min} {max} {mpd}  - Changes the default values of WRED for DSCP-based
 #random-detect dscp {value} {min} {max} {mpd}  - Changes the default values of WRED for precedence-based
```

MQC WRED (Weighted Random Early Detection)

- Used in combination with CBWFQ to prevent congestion and avoid tail-drops within a class.

COMMANDS

```
# sh policy-map interface {int}                - Shows the policy map configured with all the counters

#policy-map WRED
 #class TELNET
  #bandwidth {kbps}
  #random-detect dscp-based                    - Enables DSCP-based WRED as drop policy
  #random-detect dscp [rsvp] (value)           - Parameters for each DSCP value
 #class HTTP
  #bandwidth (kbps)
  #random-detect prec-based                    - Enables precedence-based WRED as drop policy
  #random-detect precedence [rsvp] (value)     - Parameters for each precedence value
 #class SMTP
  #bandwidth (kbps)
  #random-detect ecn                           - Enables explicit congestion notification
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-==-*
*====================*
  Shaping
*====================*
```

```
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 4: Policing and Shaping
         > Packet Flow Regulation

- Traffic-Shaping
 > Only applies to outbound traffic.
 > Queueing mechanisms can be used in conjunction with traffic shaping.
 > Traffic shaping delays packets to ensure that a class of packets does not exceed a defined rate. While delaying the packets,
      the shaping function queues the packets, by default in a FIFO queue.
 > Shaping is designed to buffer/delay traffic in excess of the configured target rate.
 > To accomplish this, a system of credits is used.
 > Before a packet can be sent a number of credits equalling the packet's size in bits must have been earned, like wages.
 > Traffic shaping does not permit the borrowing of future credits.
 > When shaping is applied to an interface, the router is given a full amount of credits. After this point all credits
      must be earned.
 > Two Types of Shaping:
      >> Generic Traffic Shaping (GTS).
      >> Frame-Relay Traffic Shaping (FRTS).
 > Two Methods of applying GTS and FRTS:
      >> Legacy method.
      >> MQC.

- Serialization/Access-Rate (AR): Physical clocking, this determines the amount of data that can be encapsulated on to the wire.
- Serialization Delay: A constant delay based on the access rate of the interface. It is the time needed to place data
      on the wire (It can't be changed).

- CIR (Committed Information Rate)
 > Dictates the required average output rate per sec on the circuit/interface.

- Tc (Time Interval)
 > The time in milliseconds into which the second is divided.
 > The Tc cannot be adjusted directly, but it can be changed by adjusting the CIR and Bc.
 > To get the Tc value correct for the formulas below, always use Tc/1000.
 > The maximum value of Tc is 125ms (1/8th of a sec) and the minimum value is 10ms (1/100th of a sec).
 > The largest amount of traffic that can be sent in a single interval is Bc + Be.
 > DO NOT use the "frame-relay tc" command to configure the Tc value, it is ONLY used for FR SVCs with a CIR=0.
 > Usually just defining an average CIR will be sufficient. But if low-latency throughput is required, changing the Tc
      might be necessary.
 > Changing the Bc value has a direct effect on the delay/time interval.

- Bc (Committed Burst)
 > Is the number of committed bits allowed to be sent per interval (Tc) to conform with the target-rate (CIR) per sec.
 > If Bc's worth of bits are sent every interval in that sec, the output rate is the CIR.
 > The Bc bucket is refilled each new Tc.
 > If there are bits left in the Bc bucket that were not used in that interval, they roll over to the Be bucket.
 > If the Be bucket is full, these excess credits are lost.
 > The Bc determines the Tc, and as a result, the amount of data to send per interval:
      >> Bigger Bc - More delay but more data per Tc.
      >> Smaller Bc - Less delay but less data per Tc (smaller Bcs are generally needed for voice).

- Be (Excess Burst)
 > Is the number of non-committed bits the router is allowed to send above the Bc if there are available credits.
 > If all the Bc per interval was not used, then at a later time the router can send Be's worth to average
      out the total amount sent up to CIR.
 > There is no time limit to how long Be can 'store' unused Bc credits. A common misconception, is that it is
      only from the previous interval.
 > Be defaults to 0-bits.

- Formulas (Tc/1000):
 > CIR = Bc / Tc.
 > Tc = Bc / CIR.
 > Bc = CIR x Tc.
 > Be = (CAR - CIR) x Tc.

*---------------------------------------*
  Legacy GTS (Generic Traffic Shaping)
*---------------------------------------*
-----------
 COMMANDS
-----------
# sh traffic-shape                               - Shows the configured shaping values per DLCI
# sh traffic-shape statistics                    - Shows packet/byte count, packets/bytes delayed

#traffic-shape {rate | group (acl)} {access-rate (bps)} [Bc (bits) [Be (bits)]] [buffer limit]
                                                 - Command syntax to enable traffic shaping on the interface
#interface s0/0
 #traffic-shape rate 640000 8000 0 1000          - AR: Configures the access rate to 64k
                                                 - Bc: The rate will not exceed 8k per time interval (Tc)
                                                 - Be: Indicates excess rate if configured. (Value of 0 here)
                                                 - Buffer-Limit is configured as 1000
#traffic-shape group 100 640000 8000 0           - All traffic matching ACL-100 will match this shaping rate
#traffic-shape fecn-adapt                        - Configures reflection of FECNs as BECNs.
#traffic-shape adaptive 32000                    - Sets the interfaces CIR at 32k. (Minimum guaranteed amount)
                                                 - If BECN received this interface will throttle to no lower than 32k
```

```
*----------------------------------------------*
  Legacy FRTS (Frame-Relay Traffic Shaping)
*----------------------------------------------*
- MINCIR
 > The rate to which the router will throttle down at a minimum, if a BECN was received from the frame-relay cloud.
 > Defaults to half the configured CIR.

- FECN (Forward Explicit Congestion Notification)
 > A message sent towards the destination to indicate congestion was experienced on the way and reflected back to the source
     as a BECN.

- BECN (Backward Explicit Congestion Notification)
 > A message sent back to the source sending the traffic as an indication to slow down the sending-rate, as there is congestion
     in the direction the traffic is being sent, but in opposite direction of the BECN.

- Adaptive Shaping
 > Used to allow the router to throttle back in the event of congestion.
 > The router will throttle back 25% per Tc when BECNs are received, and will continue to throttle back 25% each Tc until
     BECNs are no longer received or until MINCIR is reached.

- Common reasons to use FRTS:
 > To force a router to conform to the rate subscribed from the frame-relay service provider, because the local
     serialization delay is much faster than the provisioned rate, or
 > To throttle down a higher-speed site so that it does not overrun a lower speed site, typically used in partial mesh topologies.

- Be careful once FRTS is enabled on an interface:
 > All DLCIs on that interface (including sub-interfaces) are assigned the default CIR value of 56000 bps.
 > If DLCIs require a different output rate than 56k, the CIR should be adjusted.

- If FRTS is applied to a physical frame interface the config will apply to all VCs configured on that interface.
- If FRTS is applied to the VC, then the config only applies to that VC.

- Fragmentation:
 > Prevents smaller real-time packets (i.e. VOIP) from getting delayed behind big packets in the hardware FIFO queue.
!!NOTE!! The fragmentation size should be set to match the Bc, that way the worst delay = single Tc.

-----------
 COMMANDS
-----------
# sh traffic-shape                          - Shows the configured shaping values
# sh traffic-shape statistics               - Shows packet/byte count, packets/bytes delayed
# sh run map-class frame-relay FRTS         - Shows the configured map-class

#map-class frame-relay FRTS
 #frame-relay cir {bps}                     - Committed Information Rate (CIR), (default = 56000 bps)
 #frame-relay bc {bps}                      - Committed burst size (Bc), (default = 7000 bits)
 #frame-relay be {bps}                      - Excess burst size (Be), (default = 0 bits)
 #frame-relay mincir {bps}                  - Minimum acceptable CIR, (default = CIR/2 bps)
 #frame-relay adaptive-shaping becn         - Enables rate adjustment in response to BECN
 #frame-relay adaptive-shaping foresight    - Enables rate adjustment in response to foresight messages and BECN
 #frame-relay adaptive interface-congestion {queue-depth}
                                                - If the output queue depth exceeds the configured amount, slow down rate
 #frame-relay fecn-adapt                        - Enables shaping reflection of a received FECN as BECN
 #frame-relay fragment {bytes}                  - Specifies the maximum fragment size

#interface s0/0
 #frame-relay traffic-shaping                   - Step-1, Enables FRTS under the physical interface
 #frame-relay class FRTS                        - Step-2, Applies legacy FRTS to EACH VC configured on the interface OR

#interface S0/0.1
 #frame-relay interface-dlci 405
  #class FRTS                                   - Step-2, Applies FRTS only to this VC

*---------------------------------------*
  MQC CB-Shaping (Class-Based Shaping)
*---------------------------------------*
- CB-shaping is GTS applied via MQC.
- CB-shaping uses the same principles and calculations as FRTS, but does NOT adaptively shape.
- CB-shaping is supported on non frame-relay interfaces.

- Shape Average
 > Formula: Bc = shape-rate * Tc

- Shape Peak
 > Formula: shape-rate = configured-rate (1 + Be/Bc)

CONFIG-SET: Example of CB-Shape Applied to Frame-Relay Interface
+-------------------------------------------------------------------
|     policy-map FRTS-MQC
|      class class-default
|       shape average cir {bps}
|       shape max-buffers {buffer-depth}                    - Increases the buffer queue depth
|     !
|     interface s0/0
|      service-policy out FRTS-MQC                          - Normal CB-shaping applied to a frame-interface
|

*-----------------------------------------*
  MQC FRTS (Frame-Relay Traffic Shaping)
*-----------------------------------------*

CONFIG-SET: Example of FRTS applied to multipoint frame-relay interface per VC
+--------------------------------------------------------------------------
|     policy-map FRTS-MQC-R1                                - Creates a service-policy for VC going to R1
|      class class-default
|       shape average cir {bps}
|     policy-map FRTS-MQC-R2                                - Creates a service-policy for VC going to R2
|      class class-default
|       shape average cir {bps}
|       shape max-buffers {buffer-depth}                    - Increases the buffer queue depth
|      !
|     map-class frame-relay FRTS-R1
|      service-policy output FRTS-MQC                   - Calls the service-policy in the map-class
|     map-class frame-relay FRTS-R2
|      service-policy output FRTS-MQC                   - Calls the service-policy in the map-class
|      !
|     interface s0/0
|      frame map ip 10.0.0.1 501 broadcast              - Layer3-to-layer2 mapping
|      frame map ip 10.0.0.2 502 broadcast              - Layer3-to-layer2 mapping
|      frame-relay interface-dlci 501
|       class FRTS-R1                                   - Applies the class-map FRTS-R1 only to VC 501
|      frame-relay interface-dlci 502
|       class FRTS-R2                                   - Applies the class-map FRTS-R2 only to VC 502
```

Policing

- Traffic-policing is designed to drop traffic in excess of the target rate and enforce a maximum bandwidth threshold.
  > To accomplish this, a system of credits is used.
  > Before a packet can be sent a number of credits equalling the packet's size in bits must have been earned, like wages.
  > Policing differs from shaping in that the router is allowed to borrow future credits and in turn is permitted to go into a debt situation of having to "pay" back the credits.
- Policing can be applied to input or output traffic.
- Limits the rate of traffic on the interface.
- Policing is not a queueing mechanism, because traffic is not buffered for later transmission, it is either dropped or sent.

Legacy Rate-Limit (CAR)

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
    > Part 1: Classification
     > Configuring Committed Access Rate

- Uses a two-rate policer.
- If multiple statements are used on an interface, traffic will be checked top-down, but only until a statement is matched.
- Legacy CAR statement supports the 'continue' feature in having nested rate-limits, i.e. match multiple statements.
- Similarly to traffic shaping, changing the burst size determines how often the rate is enforced over a second.
- Note that rate-limit Bc/Be values are in BYTES, where shaping Bc/Be are in bits.
- Excess burst is only used when the configured Be is greater than the configured Bc.
  > For example, with Bc=1000 and Be=1000 there will be no burst.

- The Tc is typically 1.5 sec.
- Formula:
  > Bc = CIR/8 * Tc.
  > Be = Bc * 2.

```
- SYNTAX
  #rate-limit {in|output} [dscp] [access-group [rate-limit]] {CIR (bps)} {Bc (bytes)} {Be (bytes)} conform {OPTIONS} exceed {OPTIONS}

 > Input|Output                         - Defines the direction of the rate-limit statement with respect to the interface.
 > DSCP {value}                         - Allows the rate-limit to be applied to any packet matching a DSCP value.
 > access-group {acl}                   - Allows the rate-limit to be applied to any packet matching an ACL.
 > access-group rate-limit {rl-acl}     - Allows limiting of precedence groups or MAC-addresses.
 > Options:
      >> continue                       - Used to match multiple rate limit statements
      >> drop                           - Drops the packet.
      >> transmit                       - Transmits the packet.
      >> set-dscp-continue              - Sets the DSCP and continues to the next rate limit statement.
      >> set-dscp-transmit              - Sets the DSCP and sends the packet.
      >> set-prec-continue              - Sets the packet precedence and continues to the next rate limit statement.
      >> set-prec-transmit              - Sets the packet precedence and sends the packet.
      >> set-qos-continue               - Sets QOS-group and continues to the next rate limit statement.
      >> set-qos-transmit               - Sets QOS-group and sends the packet.

- Refer to the Security Chapter on how to use a "rate-limit access-list".

-----------
 COMMANDS
-----------
# sh interface {int} rate-limit                   - Shows input/output packet and byte counters

#interface s0/0
 #rate-limit input 8000 8000 8000 conform-action set-dscp-transmit 12 exceed-action set-dscp-transmit 12
                                                  - Example of how to mark ALL input traffic with DSCP-12
                                                  - This statement DOES NOT police any traffic, only MARKS
                                                  - [8000 8000 8000] arbitrary value in this case because conforming
                                                        traffic is marked with DSCP-12 and so is exceeding traffic

 #rate-limit output access-group 123 128000 24000 48000 conform-action transmit exceed-action drop
                                                  - Limits traffic matching ACL-123 to 128k

 #rate-limit output dscp 4 64000 12000 24000 conform-action transmit exceed-action drop
                                                  - Any packets sent matching DSCP-4 will be rate-limited to 64k

 #rate-limit output 192000 36000 72000 conform-action transmit exceed-action drop
                                                  - A typical "line-rate" statement
                                                  - Limits the TOTAL output rate of the interface to 192k

*-----------------*
  MQC Police
*-----------------*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 4: Policing and Shaping
         > Traffic Policing
```

```
- Uses a two- or three-rate policer, and does not support the continue feature.
- Uses an exponential formula to decide whether the formula is conforming or exceeding based on the burst rate.
- The burst value determines how often, per second, there is policing.
 > With a smaller police value, the router will police more often.
 > With a larger police value, the router will police less often.
- The Bc/Be are also configured in bytes.
- Note that although MQC police can be applied inbound/outbound on an interface, when queueing is configured in the same
      policy-map it can only be applied outbound.

- Formulas:
 > Single rate, two colour:       no violate       - Bc = CIR/32, Be = 0.
 > Single rate, three colour:     violate          - Bc = CIR/32, Be = Bc.
 > Dual rate,   three colour:     PIR              - Bc = CIR/32, Be = PIR/32.

- Options:
      >> drop                             - Drops the packet.
      >> set-discard-class-transmit       - Sets the discard-class and sends the packet.
      >> set-dscp-transmit                - Sets the DSCP and sends the packet.
      >> set-frde-transmit                - Sets the FR DE and sends the packet.
      >> set-mpls-exp-imposition-transmit - Sets the exp-bits at tag imposition and sends the packet.
      >> set-mpls-exp-topmost-transmit    - Sets exp-bits on topmost label and sends the packet.
      >> set-prec-transmit                - Rewrites the packet precedence and sends the packet.
      >> set-qos-transmit                 - Sets the QOS-group and sends the packet.
      >> transmit                         - Transmits the packet.

----------
 COMMANDS
----------
#sh int policy-map {name}
#policy-map POLICE
 #class SMTP
  #police cir 384000 bc 72000 be 144000        - CIR is in bits per second
   #conform-action {OPTIONS}                   - Bc/Be are in bytes per second
   #exceed-action {OPTIONS}
   #violate-action {OPTIONS}                   - Violate-action enables a 3-rate policer



*------------------------------------*
  COPP (Control Plane Policing)
*------------------------------------*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 4: Policing and Shaping
         > Traffic Policing
          > Control Plane Policing
```

- The COPP feature allows users to configure a QOS filter that manages the traffic flow of control plane packets to protect the control plane of routers and switches against reconnaissance and denial-of-service (DOS) attacks.
- In this way, the control plane can help maintain packet forwarding and protocol states despite an attack or heavy traffic load on the router or switch.

- Ensure that layer3 control packets have priority over other packet types that are destined for the control plane.
- The following types of layer3 packets are forwarded to the control plane:

 > Routing protocol CP (control packets).
 > Packets destined for the local IP address of the router.
 > Packets from management protocols (such as SNMP, Telnet and SSH).

- Aggregate control plane services provide COPP for all CP packets that are received from all line-card interfaces on the router.
- Distributed control plane services provide COPP for all CP packets that are received from the interfaces on a line card.

- Control-plane traffic is classified into different categories of traffic:

 > Control-plane host sub-interface
 >> Traffic which is directly destined for one of the router's interfaces.
 >> Examples of control-plane host IP traffic include tunnel termination traffic, management traffic, or routing protocols such as SSH, SNMP, BGP, OSPF and EIGRP.
 >> All host traffic terminates on and is processed by the router.
 > Control-plane transit sub-interface
 >> Traffic which is software switched by the route processor, thus packets not directly destined to the router itself but rather traffic traversing through the router.
 >> Non-terminating tunnels handled by the router are an example of this type of control-plane traffic.
 > Control-plane CEF-exception sub-interface
 >> Traffic that is either redirected as a result of a configured input feature in the CEF packet forwarding path for process switching, or directly enqueued in the control-plane input queue by the interface driver.
 >> Examples are ARP, l2 keepalives, and all non-IP host traffic.

CONFIG-SET: COPP (Control Plane Policing)

```
+------------------------------------------
|      access-list 140 permit tcp host 10.1.1.1 any eq 23    - Allows 10.1.1.1 trusted host traffic
|      access-list 140 permit tcp host 10.1.1.2 any eq 23    - Allows 10.1.1.2 trusted host traffic
|      !
|      class-map telnet-class
|       match access-group 140
|      !
|      policy-map control-plane-in
|       class telnet-class
|        police 80000 conform transmit exceed drop           - Drops all traffic that matches the Telnet-class
|      !
|      control-plane
|      service-policy output control-plane-out               - Defines the aggregate control plane service for the active RP
|      !
```

```
-----------
 COMMANDS
-----------
# sh policy-map control-plane all                 - Shows information about the all control plane policies
```

```
#control-plane [host | transit | cef | slot]     - Enters control-plane configuration mode
                                                  - [host] Applies policies to host control-plane traffic, optional
                                                  - [transit] Applies policies to transit control-plane traffic
                                                  - [cef] Applies policies to CEF-exception control-plane traffic
                                                  - [slot] Attach a QOS policy to the specified slot

 #service-policy {input|output} {p-name}          - Attaches a QOS service policy to the control plane
                                                  - {input} Applies to packets received on the control plane
                                                  - {output} Applies to packets transmitted from the control plane


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-==-*
*============================================*
   RSVP (Resource Reservation Protocol)
*============================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 5: Signaling
         > RSVP

- RSVP on its own is just a reservation tool in the control plane, it still requires an external mechanism to enforce it.
- Allows end-user application to make bandwidth reservations inside the network.
- When using "ip rsvp bandwidth" on sub-interfaces, it is also required to be configured on the main interface.
- When using multiple sub-interfaces with "ip rsvp bandwidth", the main interface should be configured to be the
      sum of all sub-interfaces.

-----------
 COMMANDS
-----------
#map-class frame-relay FRTS
 #frame fair-queue                                - WFQ required for RSVP, gets disabled by default with traffic-shape

#interface e0/0
 #ip rsvp bandwidth {int-kbps} {flow-kbps}        - Enables RSVP for IP on an interface


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-==-*
*=========================*
   AutoQOS
*=========================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 10: Autoqos

- Autoqos automates the deployment of QOS policies.
- Any existing QOS policies must be removed before the autoqos-generated polices are applied.
- Autoqos is supported only on the IP-Plus image for low-end platforms.
```

- Ensure that autoqos is enabled on both sides of the network link.
- The bandwidth on both sides of the link must be the same, otherwise a fragmentation size mismatch might occur preventing the connection from being established.
- The autoqos feature cannot be configured on a frame-relay DLCI if a map-class is attached to the DLCI.
- For frame-relay networks, fragmentation is configured using a delay of 10ms and a minimum fragment size of 60 bytes.

- Autoqos pre-requisites:

> CEF must be enabled on the interface/PVC.
> The interfaces must have IP addresses configured.
> The amount of bandwidth must be specified by using the "bandwidth" command.

- The bandwidth of the serial interface determines the speed of the link.
- The speed of the link in turn determines the configurations generated by the autoqos.
- Autoqos uses the interface bandwidth that is allocated at the time it is configured, but not after autoqos is executed.
- Autoqos for the enterprise feature consists of two configuration phases:

> Auto-Discovery (data collection)
>> Uses NBAR-based protocol discovery to detect the applications on the network and performs statistical analysis on the network traffic.
> Autoqos template generation and installation
>> This phase generates templates from the data collected during the Auto-Discovery phase and installs the templates.

- Class definitions for the enterprise autoqos:

CLASS-NAME	DSCP-VALUE	DEFAULT NBAR MATCH COMMAND
> IP Routing	CS6	bgp, ospf, eigrp, rip, rsvp, ldp.
> Interactive Voice	EF	rtp-voice, cisco-phone.
> Interactive Video	AF41	rtp-video.
> Streaming Video	CS4	vdolive, streamwork, realaudio, netshow, cuseeme.
> Telephony Signaling	CS3	rtcp, h323.
> Transactional/Interactive	AF21	sap, sql, citrix, telnet, ssh, vnc, pcanyware.
> Network Management	CS2	snmp, syslog, dns, dhcp, ldap, imap, tacacs, isakmp.
> Bulk Data	AF11	ntp, ftp, irc, tftp, pop3, smtp, netbios, cifs.
> Scavenger	CS1	napster, bittorrent, kazaa2, edonkey, gnutella.
> Best Effort	0	http, secure-http, gopher, nfs, sunrpc, ntp, rcmd & unknown.

- The "auto discovery qos" command is not supported on sub-interfaces.
- The "auto qos voip" command is not supported on sub-interfaces.

- Autoqos — VoIP (Voice of IP)

> Same as above, previous QOS policies have to be removed before running the autoqos-VoIP macro.
> All other requirements must be met too.
> The VoIP feature helps the provisioning of QOS for VoIP traffic.

```
-----------
 COMMANDS
-----------
# sh auto discovery qos [interface]          - Views the auto-discovery phase in progress, or displays the results
                                                    of the data collected
# sh auto qos [interface]                    - Shows the autoqos templates created for a specific interface or all
```

```
#interface s0/2
 #bandwidth {kpbs}                                   - Optional but always recommended
 #auto discovery qos [trust]                         - Starts the auto-discovery phase
                                                     - [trust] Indicates that the DSCP markings of packets are trusted
 #no auto discovery qos                              - Stops the Auto-Discovery phase
 #auto qos                                           - Generate the autoqos templates and installs it

#interface s0/3
 #encapsulation frame
 #bandwidth {kbps}
 #frame-relay interface-dlci 100
 #auto qos voip [trust]                              - Configures the autoqos — VoIP feature
                                                     - [trust] indicates that the DSCP markings of packets are trusted

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
  Switching QOS
*==================================*
- DOC-CD LOCATION
      > Cisco Catalyst 3560 Series Switches Configuration Guides
       > Catalyst 3560 Switch Software Configuration Guide, Rel. 12.2(25)SEE
        > Configuring QOS

- COS (Class of Service) is also known as 802.1p priority bits.
- QOS must be enabled on a switch with "mls qos".

- With "mls qos" OFF the switch does not modify any markings.
- With "mls qos" ON the switch clears all COS, ip-prec, and DCSP, unless the trust configuration was specified.

- Classification
 > If QOS is disabled globally no classification will occur.
 > To trust the incoming marking type use the command "mls qos trust".
      > For IP-traffic, ip-precedence or DSCP can be trusted.
      > For trunk links COS can be trusted,
            >> If a packet has no incoming COS or it is an access link, a default value of zero is applied.
            >> But this default value can be changed with "mls qos cos".
      > For known devices conditional trusting could be configured.
            >> Thus, only trust the COS if, for example, a cisco-phone is plugged in.
            >> Configured with: "mls qos trust device cisco-phone".
 > Alternatively, default COS classification of all incoming traffic could be forced, regardless of existing marking.
      >> Example how to override all interface traffic with COS-3:
            #interface fa0/0
             #mls qos cos override
             #mls qos cos 3

- Ingress Queueing
 > The 3560 packet scheduler uses a method called shared round-robin (SRR) to control the rates at which packets are sent.
 > On ingress queues, SRR performs sharing among the two queues according to the weights configured.
 > The weights are relative rather than absolute, i.e. percentage-based rather than bandwidth-based.
 > Firstly, specify the ratios by which to divide the ingress buffers into the two queues.
 > Configured with the command "mls qos srr-queue input buffers {percentage1} {percentage2}".
 > Then configure the bandwidth percentage for each queue, which sets the frequency at which the scheduler takes packets from the
     two buffers (even though the command says bandwidth it does NOT represent any bit rate).
 > Configured with "mls qos srr-queue input bandwidth {weight1} {weight2}".
 > These two commands determine how much data the switch can buffer before it begins dropping packets.

 > Either of the two ingress queues can be configured as a priority queue.
 > The weight parameter defines the percentage of the link's bandwidth that can be consumed by the priority queue when there
     is competing traffic in the non-priority queue.
 > The priority queue is configured with "mls qos srr-queue input priority-queue {queue-number} bandwidth {weight}".


- Egress Queueing
 > Adds a shaping feature that slows down egress traffic, which helps sub-rates for ethernet interfaces.
 > There are four egress queues per interface.
 > Queue number one can be configured as a priority/expedite queue.
 > The egress queue is determined indirectly by the internal DSCP, and the internal DSCP is compared to the DSCP-to-COS map.
 > The resulting COS being compared to the COS-to-queue map.
 > SRR on egress queues can be configured for shared mode or for shape mode.
     >> Both shared and shaped mode scheduling attempt to service the queues in proportion to their configured bandwidth
           when more than one queue holds frames.
     >> Both shared and shaped mode schedulers service the PQ as soon as possible if at first the PQ is empty but then frames
           arrive in the PQ.
     >> Both shared and shaped mode schedulers prevent the PQ from exceeding its configured bandwidth when all the other queues
           have frames waiting to be sent.
     >> The only difference in operation is that the queues in shaped mode never exceed their configured queue bandwidth setting.


- Congestion Avoidance
 > The 3560 uses WTD for congestion avoidance.
 > WTD creates three thresholds per queue into which traffic can be divided, based on COS value.
 > Tail-drop is used when the associated queue reaches a particular percentage.
 > For example, a queue can be configured so that it drops traffic with COS values of 0-3 when the queue reaches 40%,
     drops traffic with COS 4 and 5 at 60 % full, and finally drops COS 6 and 7 traffic only when the queue is 100 % full.
 > WTD is configurable separately for all six queues in the 3560 (two ingress, four egress).


- Traffic Policing
 > Can be applied to both input and output queues.
 > Two types:
     >> Individual
           >>> Applies to a single class-map like IOS.
     >> Aggregate
           >>> Applies to multiple class-maps in a single policy-map.
           >>> Classes X,Y and Z cannot exceed 640k as an aggregate.
           >>> Is applied with the global command "mls qos aggregate-policer {policy-map}".

 > A unique exceed action in the policer can be used to remark DSCP to "policed-dscp-map".
```

```
CONFIG-SET: MLS-QOS, Aggregate-Policy for HTTP and SMTP Traffic
+-----------------------------------------------------------------
|     mls qos aggregate-policer APOL 64000 8000 exceed-action policed-dcsp-transit
|     !                                               >> Step-1, Creates the aggregate policy
|     access-list 180 permit tcp any any eq 80
|     access-list 180 permit tcp any eq 80 any        - Creates a ACL to match HTTP
|     access-list 125 permit tcp any any eq 25
|     access-list 125 permit tcp any eq 25 any        - Creates a ACL to match SMTP
|     !
|     class-map HTTP
|      match access-group 180                         >> Step-2, References ACLs to match required traffic
|     class-map SMTP
|      match access-group 125
|     !
|     policy-map QOS                                  >> Step-3, Create a QOS policy-map
|      class HTTP
|       police aggreagate APOL                        - Applies the aggregate-policer to multiple classes
|      class STMP
|       police aggreagate APOL                        - Applies the aggregate-policer to multiple classes
|     !
|     mls qos                                         >> Step-4, Enables SW-QOS
|     int fa0/5
|      service-policy input QOS                       >> Step-5, Applies the policy to the interface
```

```
----------
 COMMANDS
----------
# sh mls qos                                    - Shows global QOS configuration information
# sh mls qos maps dscp-mutation [name]          - Shows the current DSCP mapping entries.
# sh mls qos maps dscp-cos                      - Shows the DSCP-to-COS map
# sh mls qos interface [buffers|queueing]       - Shows the QOS information at the port level
# sh mls qos input-queue                        - Shows the settings for the ingress queues
# sh mls qos aggregate-policer                  - Shows the QOS aggregate policer configuration

#mls qos                                        - Enables switching QOS globally
#interface fa0/1
 #mls qos vlan-based                            - Enables VLAN-based QOS on the port

#interface fa0/2
 #mls qos cos {cos}                             - Configures the default COS value for untagged packets
 #mls qos cos override                          - Enforces the COS for all packets entering the interface

#interface fa0/3
 #mls qos trust {cos|dscp|ip-prec}              - Enables trusting the incoming packet based on its marking
 #no mls qos rewrite ip dscp                    - Enables DSCP transparency. The DSCP field in the packet is left unmodified

#interface fa0/4
 #mls qos trust device cisco-phone              - Specifies that the Cisco IP Phone is a trusted device
```

```
#mls qos map dscp-cos {dscp list} to {cos}      - Modifies the DSCP-to-COS map


                                                >>> DSCP MUTATION MAP <<<
#mls qos map dscp-mutation {name} {in} to {out} - Modifies the DSCP-to-DSCP-mutation map. (default = no DSCP-to-DSCP mapping)
                                                - Maps an (up to 8) incoming DSCP values to a single outgoing DSCP value
#interface fa0/5
 #mls qos trust dscp                            - Configures the ingress port as a DSCP-trusted port
 #mls qos dscp-mutation {name}                  - Applies the mutation-map to the specified ingress DSCP-trusted port


                                                >>> INPUT QUEUE <<<
#mls qos srr-queue input buffer {rat-1} {rat-2} - Uses ratios to divides the ingress buffers into two queues
#mls qos srr-queue input bandwidth {w1} {w2}    - Configures the bandwidth percentage for each queue
#mls qos srr-queue input priority {q-no} bandwidth {weight}
                                                - Configures on ingress queue as a priority queue

                                                >>> OUTPUT QUEUE <<<
#srr-queue bandwidth share {w1} {w2} {w3} {w4}  - Assigns SRR weights to the egress queues, with share-mode
#srr-queue bandwidth shape {w1} {w2} {w3} {w4}  - Assigns SRR weights to the egress queues, with shape-mode

                                                >>> SET WTD FOR A EGRESS QUEUE-SET <<<
#mls qos queue-set output {set-id} buffers {a1}{a2}{a3}{a4}
                                                - Allocates buffers to each queue-set ID
#mls qos queue-set output {set-id} threshold {q-id} {drop-1} {drop-2} {reserve} {maximum}
                                                - Configures the WTD thresholds, guarantee the availability of buffers
#interface fa0/7
 #queue-set {set-id]                            - Maps the port to a queue-set

                                                >>> AGGREGATE POLICER <<<
#mls qos aggregate-policer {name} {rate-bps} {burst-bytes} exceed-action {drop | policed-dscp-transmit}
                                                - Defines the policer parameters to apply to multiple traffic classes
#police aggregate {name}                        - Applies the aggregate-policer to the different classes




*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Compression
*=====================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 6: Link Efficiency Mechanisms
         > Header compression

- 'Optimizing links for maximum payload throughput' is exam-speak for compression.
- If files are already compressed or in a compressed format, it is recommended to not use compression.
- TCP Header Compression
 > A mechanism that compresses the TCP header in a data packet before the packet is transmitted.
 > Configured with "ip tcp header-compression".

- STAC Compression
 > The lossless data compression mechanism is STAC, using the LZF algorithm.
 > Configured under the interface with "compress stac".

- Predictor
 > Uses the RAND compression algorithm.
 > Configured using "compress predictor" along with PPP encapsulation.

- RTP Header Compression
 > Allows the reduction of the RTP header from 40 bytes to 2-5 bytes.
 > It's best used on slow-speed links for real-time traffic with small data payloads, like VoIP.
 > To configure on a serial link use "ip rtp header-compression".
 > To enable per VC, use the command "frame-relay map ip {IP} {DLCI} [broadcast] rtp header-compression".
 > The 'passive' keyword means the router will not send RTP compressed headers unless RTP headers are received.
```

```
-----------
 COMMANDS
-----------
# sh ip tcp header-compression                  - Shows header compression statistics
# sh frame-relay map                            - Shows the configured header compression per DCLI

#interface se0/0
 #compress stac                                 - Configures lossless data compression mechanism
#interface se1/0
 #encap ppp                                     - Required for predictor
 #compress predictor                            - Enables the RAND algorithm compression

 #ip tcp header-compression                     - Enables TCP header compression

 #ip rtp header-compression [passive] [periodic-refresh]
                                                - Enables RTP header compression
                                                - [passive] Compress for destinations sending compressed RTP headers
                                                - [periodic-refresh]: Send periodic refresh packets
#interface s0/1.1
 #frame-relay map ip {ip} {dlci} rtp header-compression [connections] [passive] [periodic-refresh]
                                                - Enables RTP header compression per VC
                                                - [connections] Max number of compressed RTP connections (DEF=256)
                                                - [passive] Compress for destinations sending compressed RTP headers
                                                - [periodic-refresh]: Send periodic refresh packets
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   Troubleshooting QOS
*======================*
- When troubleshooting QOS configuration, consider the following:
 > Was the bandwidth statement used to specify the correct bandwidth amount?             # sh run int {int} | i band
 > Is the traffic classified correctly?                                                 # sh class-map
     >> Are the class-maps calling the correct ACLs?                                    # sh class-map {name}
 > If ACLs are used for classification,
     >> Does the ACL exist (matching a non-existing ACL = MATCH all traffic)?           # sh ip acce {acl}
     >> Does the ACL match the correct IPs and ranges?                                  # sh ip acce {acl}
     >> Are the ACL entries getting matches?                                            # sh ip acce {acl} | i matches
     >> If not was the ACL format correctly entered?                                    #access-list {no} permit {src} {dst}
 > Is the policy-map calling the correct class-maps?                                    # sh run policy-map
     >> Was the policy-map applied to the interface in the correct direction?           # sh run int | i service-policy
 > After applying a policy-map, confirm if all available bandwidth was allocated.       # sh int {int} | i Available
 > Does the interface show the correct queueing strategy?                               # sh int {int} | i strategy
 > Has the police-map matched any traffic?                                              # sh policy-map interface {int}
 > Has the QOS on the interface had to drop any traffic?                                # sh int {int} | i  output drops


 > With CBWFQ were the amounts specified in kbps?                                       # sh run policy-map | i bandwidth
 > With CB-shaping were the correct rate, Bc and Be values specified?                   # sh run policy-map | i shape
 > With FRTS were the correct CIR, Bc and Be values specified?                          # sh run map-class frame-relay
     >> Was traffic shaping enabled on the physical interface?                          # sh run int {int} | i shaping
 > With policing:
     >> Was the CIR specified in bps?                                                   # sh run policy-map | i police
     >> Were the Bc and Be specified in bytes?                                          # sh run policy-map | i police
     >> Was the Be configured to be larger than the Bc?                                 # sh run policy-map | i police

 > With RTP/TCP header compression, are both sides enabled?                             # sh run int {int} | i compress
     >> This is required unless 'passive' is used.
     >> Copy descretly owned by Kane  Bagwell

- How to troubleshoot whether or not packets are marked correctly:
 > Firstly enable Netflow on the interface to see CURRENT traffic flows.                #ip route-cache flow  OR  #mpls netflow
 > Have a look at the cache-flow to see the traffic (src, dst, interfaces, ports, pckts). # sh ip cache [int] flow
     >> If there are no traffic-flows, generate traffic from the source router with IP-SLA.
 > Do a verbose cache-flow to see the packets' TOS-byte values on arrival (look at TOS).  # sh ip cache [int] ver flow
     NOTE: The cache-flow is taken BEFORE any packet markings! Local marking WILL NOT show.

 > Know how to calculate the TOS-byte HEX value to DSCP PHB or DSCP decimal value.
 > Let's use a TOS-byte of 48:
     >> Convert 48 from hex to binary to get the 8-bit breakdown: 48 = 01001000
     >> Since we are only interested in the first 6 bits that make up the DSCP value, remove the last two zeros.
     >> Convert the remaining 6 bits to decimal. 010010 -> 18.
     >> Thus 48-HEX provides a DSCP decimal value of 18 or DSCP AF21.
 > Don't forget that locally marked DSCP values can't  be seen in a local "ip cache-flow", it is only visible one hop further on.
 > If you want to work out the decimal value for a DSCP AFxy value, use the formula (8x + 2y).
      >> Example AF31:
           = (8*x) + (2*y)
           = (8*3) + (2*1)
           = AF31 = 26.




*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-==-*
*====================*
     OUTPUT-101
*====================*
--------------------------------------------------------
     # sh policy-map interface s0/1/0
--------------------------------------------------------
  Service-policy output: SHAPE
    Class-map: class-default (match-any)

      Match: any
      Traffic Shaping
           Target/Average   Byte   Sustain   Excess    Interval Increment
             Rate           Limit  bits/int  bits/int  (ms)       (bytes)
            64000/64000     2000   8000      8000      125        1000

        Adapt  Queue     Packets   Bytes     Packets   Bytes     Shaping
        Active Depth                         Delayed   Delayed   Active
        -       0        0         0         0         0         no

- Target Rate                  = CIR
- Byte-Limit                   = Bc+Be ie the size the token bucket, express in BYTES
- Sustain bits/int             = Bc value, (int is short for interval or Tc)
- Excess bits/int              = Be value
- Interval (ms)                = Tc value
- Increment (bytes)            = How many bytes of token replenished each Tc, i.e. Bc value in bytes
- Adapt Active                 = Shows adaptive shaping has been enabled. If a BECN is received, the flow will be throttled back.
```

```
  ______                     _
 / _____)                   (_)  _
( (____  _____  ____ _   _  ____ _ _| |_ _   _
 \____ \| ___ |/ ___) | | |/ ___) (_   _) | | |
 _____) ) (___( (___| |_| | |   | | | |_| |_| |
(______/\_____) \____)____/|_|   |_|  \__)\__  |
                                        (____/
*-=-=-=-=-=-=-=-=-=-=-=-*
|      INDEX            |
*-=-=-=-=-=-=-=-=-=-=-=-*
```

```
- ACLs (Access Control Lists)
      + Standard ACLs
      + Extended ACLs
      + Named ACLs
      + Extended ACLs and IGPs
      + Extended ACLs for BGP Filtering
      + Scott Morris' Binary Math for ACLs
      + ACL Logging
      + Applying ACLs
- Rate-Limit ACLs
- Time-Based ACLs
- Dynamic ACLs
- Reflexive ACLs
- CBAC (Content-Based Access-Control)
- ZBFW (Zone-Based policy Firewall)
- IPS (Intrusion Prevention Systems)
- Common Number Ranges
      + Port Numbers
      + Port Ranges
      + Protocol Numbers
- Security Compliance RFCs
      + RFC 1918
      + RFC 3330
      + RFC 2827
- TCP Intercept
- IP Source Tracking
- IP Traffic Export
- URPF
- Device Security Configuration
      + Passwords
      + Privilege Levels
      + CLI Session Protection
- AAA (Just basics)
      + Modes
      + Defaults
      + RADIUS/TACACS
      + Max-Attempts
      + Quiet Period
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*================================*
   ACLs (Access Control Lists)
*================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4T
          > Access Control Lists (ACLs)

- ACLs allow a router to filter network traffic based on a variety of criterias.
- ACLs are used by almost every technology in this book.
- An ACL is configured in global configuration mode, but is applied at the interface level.
- An ACL does not take effect until it is applied to an interface with the "ip access-group" command.
- Packets can be filtered as they enter or before they exit an interface.

- An ACL consists of lines of criteria that allows or denies traffic. Each line called an ACE (Access Control Entry).
- If a packet enters or exits an interface with an ACL applied, the packet is compared against the criteria of the ACE.
- The ACL is executed from the top to the bottom, ACE after ACE, until the packet matches an ACE's criteria.
- The matching ACE's action, 'permit' or 'deny', will determine if that packet is forwarded or dropped.
- If the packet did not match any of ACEs configured in the ACL, the packet will be dropped.
- At the end of every ACL is an 'implicit deny', that will drop ALL traffic not matched by any ACEs.

- To understand how ACLs operate, a good understanding of binary and subnet masks are required.
- Masks are used with IP addresses in ACLs to specify specific criteria.
- ACL use what is called an inverse mask or wildcard mask (reverse of a subnet mask i.e. 0.0.0.255).
- Subtract the normal subnet mask from 255.255.255.255 in order to determine the ACL inverse mask.
- The value of the inverse mask is broken down into binary (0s and 1s), the result determines which address bits are
      to be considered in processing the traffic.
- A 0 indicates that the address bits must match exactly; a 1 indicates that the address bits does not have to match.
- Example: Convert the following to binary: 10.1.1.0 0.0.0.255:
      >> Network (binary)          - 00001010.00000001.00000001.00000000
      >> Mask (binary)             - 00000000.00000000.00000000.11111111
      >> Thus the first three octets '10.1.1' must match a given address, where the last octet may be any value.

- More ACL examples:
      >> 10.1.0.0 0.0.255.255      - Matches 10.1.0.0/16 (from 10.1.0.0 to 10.1.255.255).
      >> 10.1.1.64 0.0.0.15        - Matches 10.1.1.64/28 (from 10.1.1.64 to 10.1.1.79).
      >> 10.1.1.2 0.0.0.0          - Matches the host 10.1.1.2.
      >> 0.0.0.0 255.255.255.255   - Matches "any".

- The router uses the terms in, out, source and destination as references in ACLs:
 > Src (Source)            - The address of the device/router that sent the packet.
 > Dst (Destination)       - The address of the device/router that a packet is destined to.
 > In                      - Traffic that arrives on an interface before it goes through the router.
 > Out                     - Traffic that has already been through the router, before it leaves the interface.

!!NOTE!! To debug traffic a packet level, interface fast-switching must be disabled with "no ip route-cache".

- There are different types of ACLs available to match different criterias.
```

```
*---------------------*
   Standard ACLs
*---------------------*
- Control traffic by the comparison of the source IP address to the addresses configured in the ACL.
- Numbered standard ACL ranges: 1-99, 1300-1999.
- SYNTAX:
      #access-list {number} {permit|deny} {src host|wildcard|any}.


*---------------------*
   Extended ACLs
*---------------------*
- Control traffic by the comparison of the source and destination IP addresses to the addresses configured
      in the ACL.
- Numbered extended ACL ranges: 100-199, 2000-2699.
- SYNTAX:
      #access-list {number} {permit|deny} ip {src host/wildcard} {dst host/wildcard} [options]
      #access-list {number} {permit|deny} {udp|tcp} {src host/wildcard} [port] {dst host/wildcard} [port] [options]
      #access-list {number} {permit|deny} icmp {src host/wildcard} [icmp-type] {dst host/wildcard} [icmp-type]

- Can match any of the following:
      > IP protocol number.
      > SRC/DST address.
      > TCP/UDP ports (eq, neq, lt gt range).
      > ICMP type codes.
      > Packets marking (DSCP, IP Precedence, TOS).
      > Time-ranges.

- To allow ICMP (refer to ICMP Section for full details):
      #permit icmp any any echo                 - Permits ping packets
      #permit icmp any any echo-reply           - Permits ping replies
      #permit icmp any any time-exceeded        - Traceroute: Permits each hop to respond when the TTL=0
      #permit icmp any any port-unreachable     - Traceroute: Permit final hop to respond


*----------------------*
   Named ACLs
*----------------------*
- Allows the options to give Standard and Extended ACL names.
- Named ACLs configuration mode numbers and sequences the ACEs, allowing easier ACL editing.
- (Recent IOS version allow this for numbered ACLs too)
- SYNTAX :
      #ip access-list {extended|standard} name
       #[seq] {permit|deny} ip {src host/wildcard} {dst host/wildcard} [options]
       #[seq] {permit|deny} {udp|tcp} {src host/wildcard} [port] {dst host/wildcard} [port] [options]
```

```
*------------------------------*
   Extended ACLs and IGPs
*------------------------------*
```

- SOURCE: http://blog.internetworkexpert.com/2008/01/04/using-extended-access-lists-in-a-distribute-list.
- Extended ACLs can be used with IGP protocols to match the network portion of the route and the IP address of the router that sent the route.
- '0' means exact match & '255' means any match.

- SYNTAX:

```
     #access-list {number} {permit|deny} [route-source] [network]
```

- EXAMPLES:

```
 > This would permit any 10.X.X.X/X network from 1.1.1.1 (i.e. 10.5.0.0/16, 10.1.1.4/30, 10.50.6.128/25, 10.1.1.64/26, etc.)
      #access-list 100 permit ip host 1.1.1.1 10.0.0.0 0.255.255.255

 > This would permit any 10.1.X.X/X network from 1.1.1.1 (i.e. 10.1.1.0/24, 10.1.5.4/30, 10.1.50.128/25, 10.1.3.64/26, etc.)
      #access-list 100 permit ip host 1.1.1.1 10.1.0.0 0.0.255.255

 > This would permit any 10.1.1.X/X network from 1.1.1.1 (i.e. 10.1.1.0/24, 10.1.1.0/30, 10.1.1.128/25, 10.1.1.64/26, etc.)
      #access-list 100 permit ip host 1.1.1.1 10.1.1.0 0.0.0.255

 > A wild card mask could also be used on the host:
 > This would permit any 10.X.X.X/X network from 1.1.1.X (i.e. 10.5.0.0/16, 10.1.1.4/30, 10.50.6.128/25, 10.1.1.64/26, etc.)
      #access-list 100 permit ip 1.1.1.0 0.0.0.255 10.0.0.0 0.255.255.255
```

CONFIG-SET: EXT-ACL to match a network from a host with a distribute-list

```
+-----------------------------------------------------------------------------------------
BEFORE:
|     R1#show ip route rip
|      R    176.16.0.0/16 [120/1] via 10.0.0.3, 00:00:06, Ethernet0/0
|                         [120/1] via 10.0.0.2, 00:00:06, Ethernet0/0
|
CONFIG:
|     access-list 100 deny ip host 10.0.0.3 host 176.16.0.0       - Matches 176.16.0.0 prefix from host 10.0.0.3
|     access-list 100 per ip any any
|     router rip
|      distribute-list 100 in e0/0
|
AFTER:
|     R1#show ip route rip
|      R    176.16.0.0/16 [120/1] via 10.0.0.2, 00:00:02, Ethernet0/0
|
```

```
*-------------------------------------*
   Extended ACLs for BGP Filtering
*-------------------------------------*
```

- SOURCE: http://blog.internetworkexpert.com/2008/01/08/using-extended-acls-for-bgp-filtering.
- Prior to the support of prefix-lists in the IOS, advanced filtering for BGP was done using extended ACLs.
- The source portion of the extended ACL is used to match the network portion of the BGP route and the destination portion of the ACL is used to match the subnet mask of the BGP route.

- SYNTAX:

```
     #access-list {no} {permit/deny} ip [network] [mask] [prefix-mask] [mask]
```

- EXAMPLES:

> Matches only 10.0.0.0/16.

```
     #access-list 100 permit ip 10.0.0.0 0.0.0.0 255.255.0.0 0.0.0.0
```

> Matches only 10.1.1.0/24 .

```
     #access-list 100 permit ip 10.1.1.0 0.0.0.0 255.255.255.0 0.0.0.0
```

> Matches 10.0.X.0/24 - Any number in the third octet of the network with a /24 subnet mask.

```
     #access-list 100 permit ip 10.0.0.0 0.0.255.0 255.255.255.0 0.0.0.0
```

> Matches 10.X.X.X/28 - Any number in the second, third & fourth octets of the network with a /28 subnet mask.

```
     #access-list 100 permit ip 10.0.0.0 0.255.255.255 255.255.255.240 0.0.0.0
```

> Matches 10.X.X.X/24 to 10.X.X.X/32 - Any number in the second, third & fourth octets of the network with a /24 to /32 subnet mask.

```
     #access-list 100 permit ip 10.0.0.0 0.255.255.255 255.255.255.0 0.0.0.255
```

> Matches 10.X.X.X/25 to 10.X.X.X/32 - Any number in the second, third & fourth octets of the network with a /25 to /32 subnet mask.

```
     #access-list 100 permit ip 10.0.0.0 0.255.255.255 255.255.255.128 0.0.0.127
```

Scott Morris' Binary Math for ACLs

- SOURCE: http://blog.internetworkexpert.com/2008/09/15/binary-math-part-i
- SOURCE: http://blog.internetworkexpert.com/2008/11/03/binary-math-part-ii
- For dissimilar networks where the shortest possible ACLs (to include only what is asked for) must be created, follow these steps:

1- Convert the octet in question to the binary of each address and find similarities.
2- Compare the bits between each number to form an ACL binary mask, '0' = all the same & '1' = differences.
3- Confirm the possibilities of the smallest possible amount, with (2^x) where x = number of 1 in the mask.
4- Convert the binary mask to decimal to get the value of the octet in question.

- EXAMPLE STEPS: Create a one line ACL to match both of these networks: 168.208.3.0/24 and 168.192.3.0/24.

1- Convert second octets in question to binary:

```
     192   11000000
     208   11010000
```

2- MASK &gt; 00010000 = Only the fourth bit differs between the two.

3- Possibilities, 1-bit difference = 2^1 = 2 possibilities. Smallest &gt; CHECK.

4- Convert to decimal: 00010000 = 16, thus the solution

```
     #access-list 11 permit 168.192.3.0 0.16.0.0
```

```
- EXAMPLES:
 > Permit all EVEN /24s in the third octet for prefix 192.168.0.0.
      #access-list 12 permit 192.168.0.0 0.0.254.0

 > Permit all ODD /24s in the third octet for prefix 192.168.0.0.
      #access-list 13 permit 192.168.1.0 0.0.254.0

 > Allow packets from all hosts in every fourth /24 network from 131.102.0.0/16.
      #access-list 16 permit 131.102.0.0 0.0.252.255

 > Match all networks with even numbers in the third octet, from 128-135 for 200.100.128.0/24.
      #access-list 17 permit 200.100.128.0 0.0.6.0

 > Match only traffic from even-numbered hosts in the second-half of the IP range 150.100.32.0/24.
      #access-list 18 permit 150.100.32.128 0.0.0.126

*-------------------*
   ACL Logging
*-------------------*
- The ACL history can be logged to console, monitor, buffer, or syslog.
- Log Options:
      >> List name/number.
      >> permit/deny.
      >> Protocol name/number.
      >> Src/Dst IP.
      >> Port number.
- The "log-input" includes the log options, the source layer2 MAC address and the input interface.

- Addition Logging Options:
 > Logging interval
      >> The interval configured in the command allows only one packet per interval to be process switched no matter how
             many log-enabled ACEs exist.
 > Logging Threshold
      >> Defines how often syslog messages are generated and sent after the initial packet match.
      >> Log messages are sent at the first matching packet and at five-minute intervals thereafter.

 > Logging Rate-Limit of Syslog Messages
      >> Limits the CPU impact of log generation and transmission.
      >> Applies to all syslog messages.
      >> Limits the number of packets that must be generated and sent by the network logging device.
      >> It does nothing to reduce the number of input packets that are process switched by the device CPU.

*------------------------*
   Applying ACLs
*------------------------*
- ACLs are applied to interfaces with "ip access-group".
- Only one ACL per interface, per direction, per protocol is allowed.
- ACLs can also be to control exec access with "access-class".
!!NOTE!! Local Traffic generated by a router does not pass through a interface's 'out' ACL. (refer to config-set below)
```

```
CONFIG-SET: Policy Route Local Router Traffic via an ACL
+----------------------------------------------------------
|Allows locally generated router traffic to 're-enter' the router and be passed through an ACL
|
|     ip access-list extended LOCAL_TRAFFIC
|      permit tcp any any eq 23                             - Matches locally-generated Telnet traffic
|     !
|     route-map LOCAL_POLICY 10
|      match ip address LOCAL_TRAFFIC                       - Redirect local Telnet traffic via the loopback interface
|      set interface Loopback0                              - Traffic sent to loopback interface 're-enters' the router
|     !
|     interface loopback0
|      ip address 150.1.6.6 255.255.255.50
|     !
|     ip local policy route-map LOCAL_POLICY                - Applies the local policy

-----------
 COMMANDS
-----------
# sh ip access-list                             - Shows the contents of all current ACL
# sh ip access-list {number|name}               - Shows the contents of the specified ACL
# sh logging                                    - Shows the console buffer

# terminal monitor                              - Shows logging output to the current terminal line
# terminal no monitor                           - Turn the display logging to the terminal monitor off

#logging monitor [level]                        - Enables terminal line (monitor) logging parameters
#logging console [level]                        - Enables console logging parameters
#logging buffered [size] [level]                - Enables logging to the buffer

#access-list 1 permit 10.1.1.0 0.0.0.255        - Permits traffic from the source range 10.1.1.0/24
#access-list 101 permit tcp host 1.1.1.1 any eq www log-input
                                                - Matches www traffic and logs the allowed traffic and source interface
#access-list 101 permit icmp any any echo-reply - Permit ping replies
#access-list 101 permit udp any gt 1023 any gt 1023
                                                - Allows reply high port traffic such as TFTP
#access-list 101 deny tcp any any log-input     - By specifying TCP/UDP/ICMP opposed to just IP, provides greater detail
#access-list 101 deny udp any any log-input            in the logging buffer, e.g. the port numbers etc
#interface fa0/0
 #ip access-group 1 in                          - Applies the standard ACL in an inbound direction on the interface
 #ip access-group 101 out                       - Applies the extended ACL in an outbound direction on the interface

#line vty 0 15
 #access-class 1 in                             - Applies the standard ACL to incoming VTY traffic

#ip access-list logging interval {msec}         - Specifies every 'ms' a log entry is create when a match occurs
#ip access-list log-update threshold {#hits}    - Specifies how many ACL hits before generating the log entry
                                                - Defaults to the first matching packet and at 5-min intervals thereafter
#logging rate-limit {msg-rate}  [except severity-level]
                                                - Limits the number of syslog messages created
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
   Rate-Limit ACLs
*============================*
- Can be used to limit traffic based on MAC addresses, IP precedence or MPLS EXP bit values.
- The matching of MAC addresses and a single IP precedence value is straight forward.

- Matching Multiple Values
 > IP precedence or MPLS EXP bit values can be matched in a single entry by using the 'mask' function.
 > To use the mask option understand the following first:
      >> There are eight IP precedence or EXP bit values 0-7.
      >> The possible binary values are:
            >>>  0      = 00000001
            >>>  1      = 00000010
            >>>  2      = 00000100
            >>>  3      = 00001000
            >>>  4      = 00010000
            >>>  5      = 00100000
            >>>  6      = 01000000
            >>>  7      = 10000000

      >> To get the binary vector add the binary values of the required IP precedence or EXP bit value together.
      >> The binary bit vector consists of the eight IP precedence values: [p7][p6][p5][p4][p3][p2][p1][p0].
      >> Let's assume IP precedence 1, 3 and 7 must be matched.
      >> Add the binary values of 1, 3 and 7 together to get 10001010.
      >> Then convert 10001010 to hex to get the mask value:
            > Convert the first four bits to decimal and then the last four bits.
            > Then convert each decimal value to hex.
            > For example:    1000 = 8 and 1010 = 10.
            > 10 in decimal is equal to A in HEX.
            > Thus 10001010 provides a rate-limit mask of 8A.

CONFIG-SET: Different Kinds of Rate-Limit Statements
+---------------------------------------------------------
|     access-list rate-limit 10 2                      - Matches IP precedence 2 only
|     access-list rate-limit 11 mask 8A                - Matches IP precedence 1,3 and 7
|     access-list rate-limit 112 c200.2bcb.0000        - Matches a MAC address
|     !
|     interface fa0/1
|      rate-limit input dscp 36 64000 12000 24000 conform transmit exceed drop
|                                                      - Limits all DSCP AF42 traffic to 64k
|
|      rate-limit input access-group rate-limit 10 64000 12000 24000 conform transmit exceed drop
|                                                      - Limits IP precedence 2 traffic to 64k
|
|      rate-limit input access-group rate-limit 11 64000 12000 24000 conform transmit exceed drop
|                                                      - Limits IP precedence 1,3 and 7 traffic to 64k
|
|      rate-limit input access-group rate-limit 112 64000 12000 24000 conform transmit exceed drop
|                                                      - Limits all traffic from the MAC to 64k
```

```
----------
 COMMANDS
----------
# sh int {int} rate-limit                      - Shows the interface rate-limit counters

#access-list rate-limit {0-99} {prec | mask {hex-value}}
                                               - Creates a IP Precedence rate-limit ACL
#access-list rate-limit {100-199} {MMMM.AAAA.CCCC}
                                               - Creates a MAC address rate-limit ACL
#access-list rate-limit {200-299} {EXP | mask {hex-value}}
                                               - Creates a EXP bit rate-limit ACL
#interface fa0/1
 #rate-limit {in|output} [dscp] [access-group [rate-limit]] {CIR} {Bc} {Be} conform {OPTIONS} exceed {OPTIONS}
                                               - Configures the required rate-limit


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   Time-Based ACLs
*======================*
- Similar to extended ACLs in function, Time-Based ACLs allow for access control based on time.
- A time range is created that defines specific times of the day and week in order to implement time-based ACLs.
- The time range is identified by a name and then referenced by a function.
- The time restrictions are imposed on the function itself. The time range relies on the router system clock.

CONFIG-SET: Timed-Based ACL Example
+-------------------------------------------
|     time-range OFFICE                                 - Creates the time-range group
|      periodic weekdays 9:00 to 16:59                  - Specifies the allowed times
|     !
|     ip access-list ext TIMEBASED
|      10 permit tcp any any eq www time-range OFFICE   - References the time-range group in the ACL
|      20 deny tcp any any eq www                       - Denies web traffic outside the permitted time window
|      30 permit ip any any
|     !
>      Extended IP access list TIMEBASED                - Shows IP access-list output
>        10 permit tcp any any eq www time-range OFFICE (inactive)
>                                                       - Will show the time-window status, 'active' or 'inactive'


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   Dynamic ACLs
*======================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4
          > Configuring Lock-and-Key Security (Dynamic Access Lists)
```

```
- Also known as 'Lock and Key' ACLs
- Application example from the inside:
      >> User must first authenticate, before being allowed to send traffic to the internet.
      >> Poor man's version of configuration proxy.

- Application example from the outside:
      >> User must first authenticate, before being allowed to access internal web server.

- Typically, there are two ACL formats (there must be at least ONE DYNAMIC PERMIT entry above a deny statement):
      >> With the EXPLICIT permit format:
            1- Dynamic permit
            2- Static deny
            3- Explicit permit
      >> With the IMPLICIT deny format:
            1- Dynamic permit
            2- Implicit deny
```

CONFIG-SET: Dynamic ACL - Creating and Applying

```
+------------------------------------------------------
|     access-list 100 permit tcp any host 195.1.0.5 eq 23         - Explicitly permits a host to Telnet into the local router
|     access-list 100 dynamic MY-DYN-ACL permit tcp any any eq 25 - Specifies the dynamic entry
|     access-list 100 deny tcp any any eq 25                      - Denies all unauthenticated traffic
|     access-list 100 permit ip any any log-input                 - Allows all other traffic
|     !
|     interface s0/0
|      ip access-group 100 in                                     - Applies ACL-100 to the interface
|     !
>     sh ip acce 100                                              - Shows the static and dynamic entries
>     clear access-template 100 MY-DYN-ACL host 195.1.0.3 any     - Clears the dynamically-created entry
```

```
- To authenticate and test Lock-and-Key
 > Telnet to the lock-and-key router.
 > Authenticate with username and password.
 > See the config-set below for three different methods to create activation ACL entry.
 > If successful, dynamic-ACL entry will be created.
 > Then test connectivity to the destination device located behind the lock-and-key router,
      e.g. "telnet 195.1.15.3 25".
```

CONFIG-SET: Dynamic ACL - Activation can be achieved three ways

```
+------------------------------------------------------
|1st -
|     username BOB password CISCO                           - Configures per-user based authentication
|     username BOB autocommand access-enable [host]         - Activate the dynamic-ACL when username BOB successfully logs in
|                                                           - [host] Create the dynamic entry based on source address
|2nd -
|     line vty 0 4
|      autocommand access-enable [host]                     - Same as method-1, but applies to all local access connections
|                                                           - [host] Create the dynamic entry based on source address
|      autocommand-options [nohangup]                       - Disables default behavior of disconnecting a user after auth
```

```
|3rd -
|    router> access-enable                                 - Once successfully authenticated, issue the command manually
|                                                               on lock-and key router to allow access


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=======================*
   Reflexive ACLs
*=======================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4
          > Configuring IP Session Filtering (Reflexive Access Lists)

- Also known as 'IP session filtering'.
- An ACL used to track outbound traffic by dynamically allowing return inbound traffic, based on the outbound traffic flows.
- Any traffic where the return traffic is not a mirror of the outgoing traffic won't work and has to be manually allowed.
- Outbound access-lists do not match locally router-generated traffic, like routing protocols, which must be manually permitted.
- By statically permitting traffic outbound, you are also required to allow the traffic back in.

CONFIG-SET: Reflexive ACL Example
+---------------------------------------
|     ip access-list extended OUTBOUND                            - Creates the outbound ACL
|      permit icmp any any reflect STATEFUL
|      permit tcp any any reflect STATEFUL                        - Specifies what traffic needs to be reflected
|      permit udp any any reflect STATEFUL
|     !
|     ip access-list extended INBOUND                             - Creates the outbound ACL
|      permit icmp any any echo-reply                             - Have to manually allow ping replies
|      permit icmp any any time-exceeded                          - Have to manually allow trace to complete
|      permit icmp any any port-unreachable                       - Have to manually allow trace to complete
|      permit tcp any any eq bgp
|      permit tcp any eq bgp any                                  - Have to manually allow routing protocol traffic
|      permit eigrp any any                                       - Also remember to allow local-router traffic back in!!!
|      permit udp any any eq 520                                  - Allows RIP traffic
|      evaluate STATEFUL                                           - This creates the dynamic reflect ACL-entries
|

-----------
 COMMANDS
-----------
#ip reflexive-list timeout {seconds}          - Changes global timeout value for temporary reflexive ACEs
#ip access-list extended {name}               - If applied on an external interface, use an outbound ACL or
                                              - If applied on an internal interface, use an inbound ACL
 #permit {prot} {ip} {ip} reflect {name} [time] - Defines the reflexive access list using the reflexive permit entry
 #evaluate {rname}                            - Creates the dynamic reflect ACEs

#int e0/0
 #ip access-group {name} {in|out}             - Applies the extended access list to the interface's traffic
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*======================================*
   CBAC (Content Based Access-Control)
*======================================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4
          > Content Based Access-Control

- Intelligently filters TCP and UDP packets based on application-layer protocol session information.
- Can be configured to permit specific TCP and UDP traffic through a firewall, only when the connection is initiated from
      within the network you want to protect.
- Examines not only network-layer and transport-layer information, but also examines the application-layer protocol
      information (such as FTP connection information), to learn about the state of the session.
- Inspects traffic that travels through the IOS firewall to discover and manage state information for TCP and UDP sessions.
- This state information is used to create temporary openings in the firewall's access lists to allow return traffic and
      additional data connections for permissible sessions.
- Has the ability to detect and prevent certain types of DOS attacks, such as SYN-flooding.
- Available only for IP protocol traffic. Only TCP and UDP packets are inspected.

- If you have an outbound IP access list at the external interface, the access-list can be a standard or extended access list.
- This outbound access-list should permit the desired traffic to be inspected by CBAC.
- If traffic is not permitted, it will not be inspected by CBAC. It will simply be dropped.

- The inbound IP access-list at the external interface must be an extended access-list.
- This inbound access-list should deny the desired traffic to be inspected by CBAC.
- CBAC will create temporary openings in this inbound access-list as appropriate to permit only return traffic that is part of
      a valid, existing session.


CONFIG-SET: CBAC (Content Based Access-Control) Example
+-----------------------------------------------------------
|     ip inspect name CBAC udp                             - Configures the protocol specified to be inspected
|     ip inspect name CBAC tcp
|     ip inspect name CBAC icmp
|     !
|     ip access-list ext INBOUND
|      permit icmp any host 185.1.1.1 echo-reply           - Explicitly allows the local router to ping out fa0/0
|      permit tcp any any eq bgp                                 and receive replies
|      permit tcp eq bgp any                               - Have to manually allow routing protocol traffic
|      deny ip any any                                     - No explicitly-permitted traffic will be inspected
|     !
|     int fa0/0
|      ip access-group INBOUND in                          - Manually allow traffic to originate from outside
|      ip inspect CBAC out                                 - This will inspect outbound traffic and create the dynamic
|                                                                ACL entries inbound at the top of the inbound ACL
```

```
-----------
 COMMANDS
-----------
# sh ip inspect name {NAME}                    - Shows a particular configured inspection rule
# sh ip inspect config                         - Shows the complete CBAC inspection configuration

#ip inspect name {NAME} {prot}                 - Configures CBAC inspection for an application-layer protocol
#ip inspect name {NAME} tcp [audit][time]      - Enables CBAC inspection for TCP packets
#ip inspect name {NAME} udp [audit][time]      - Enables CBAC inspection for UDP packets
#ip inspect name {NAME} {prot} audit-trail on  - Enables audit trail for a specific protocol

#ip inspect audit-trail                        - Turns on CBAC audit trail messages

#int fa0/0
 #ip inspect {NAME} {in | out}                 - Applies an inspection rule to an interface
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================================*
   ZBFW (Zone-Based Policy Firewall)
*=====================================*
- DOC-CD LOCATION
      > Cisco IOS Software Releases 12.4T
       > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4T
        > Zone-Based Policy Firewall

- The ZBFW utilizes CBAC technology, but provides additional functionality.
- Typically meant for deployment in branch offices.

- Features
 > Stateful firewall, layer3 through layer7 with deep packet inspection.
 > Dynamic protocol and application engines for seamless granular control.
 > Application inspection and control, visible into both control and data channels to help ensure protocols and
       application conformance.
 > URL-filtering.
 > VRF-aware.
 > Supports all interfaces types.
 > Virtual firewall provides separation between virtual contexts and overlapping IP addresses.
 > Transparent layer2 firewall Can be deployed in existing networks without changing the statically defined IP addresses.
 > Resiliency - High availability for users and applications with stateful firewall failover.

- Security Zones
 > Allow grouping of physical and virtual interfaces into security zones.
 > Firewall policies are applied to traffic traversing zones, not interfaces.
 > An interface can be assigned to only one security zone.
 > By default, traffic is permitted between interfaces belonging to the same security zone.
 > By default, traffic is blocked between interfaces from different zones.
 > Traffic between an interface in a security zone and an interface not in a security zone is blocked.
 > Zones are configured with the command "zone-member security".
```

```
- Zone-Pairs
 > A zone-pair allows a unidirectional firewall policy to be specified between two security zones.
 > To allow traffic between zones, a zone-pair must be defined and a direction inspection policy must be applied to
       that pair (source-zone, destination-zone).
 > Configured with the command "zone-pair security (name) source-zone destination-zone".

- SELF-Zone
 > There is a default zone, called 'self' with a router's own IP address.
 > Traffic to and from the self-zone is permitted by default for management and control plane traffic.
 > An explicit policy can be configured to change this behavior for traffic originated by the router.
 > Take care when doing the above; remember to allow protocol traffic, as there is a default DROP-ANY in a policy-map.
 > Limited functionality available for self-zone compared to interzone traffic.
 > The stateful inspection allowed is for router-generated traffic only: TCP, UDP, ICMP & H.323.
 > Inspection for HTTP, FTP etc. is NOT available.
 > Session-and rate-limiting cannot be configured on self-zone policies.

- Class-Maps
 > Type can be match-all (AND logic) or match-any (OR logic) (same MQC QOS).
 > Matching options are ACLs and the "match protocol" command (protocols supported are the same as CBAC).
 > May combine both ACL and protocol-matching commands, but NOT multiple protocol-matching commands and ACL matching.
 > If multiple match-protocol commands are needed along with ACL matching, nested class-maps with "match class-map NAME"
      must be used.

- Policy-Maps
 > With ZBFW, there are three policy actions under the inspect-type policy-maps:
      >> Inspect  - Allows stateful inspection of traffic, from source to destination and automatically permits returning traffic.
                  - If using the inspect option, the referenced class-map MUST have at least one 'match protocol', to specify
                       the protocols to be inspected, otherwise all the protocols will be inspected.
      >> Drop     - Silently discards matching packet flows.
      >> Pass     - Permit/allow traffic WITHOUT stateful inspection.
                  - Return traffic MUST be manually allowed.

- ZBFW Rate-Limiting
 > Traffic exceeding traffic bursts will be dropped. There is NO remarking option available.
 > There is no optimal value for the burst parameter.
 > A smaller burst causes less traffic to be sent the instant after an idle period.
 > A larger burst ensures smoother traffic flow but at the risk of possible heaving traffic burst spikes.
 > ZBFW supports two types of rate-limiting:
      1- Limiting aggregate packet rate for the flows between security zones.
      2- Limiting the maximum number and/or rate of the half-open connections for TCP/UDP sessions.
            >> This is applied via "inspect parameter-map".

- Parameter-Maps
 > A parameter map allows one to specify parameters that control the behavior of actions and match criteria specified under
      a policy-map and a class-map, respectively.
 > There are currently three types of parameter maps:
      1- Inspect parameter-map
            >> An inspect parameter-map is optional.
            >> If one does not configure a parameter map, the software uses default parameters.
            >> Parameters associated with the inspect action apply to all nested actions (if any).
```

2- URL filter parameter map
>> A parameter-map is required for URL filtering.
3- Protocol-specific parameter map
>> A parameter map is required for an instant messenger application (layer7) policy map.

- Port-Mapping

> DOC-CD LOCATION
> > 12.4T Configuration Guides
> > > Security and VPN
> > > > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4T
> > > > > Configuring Port to Application Mapping

> Aka PAM (Port-Application-Mapping)
> Network applications that use non-standard ports require user-defined entries in the mapping table.
> The "ip port-map" command associates TCP or UDP port numbers with applications or services, establishing a table of default port mapping information at the firewall.
> These entries automatically appear as an option for the "ip inspect name" command to facilitate the creation of inspection rules.
> If a well-known port needs to be changed for a different application, the 'list' keyword referencing an ACL must be used.
> Example : Here Real-Audio is using port-21 usually reserved for FTP-control.

```
#access-list 10 permit 192.168.32.43
#ip port-map realaudio port 21 list 10
```

- ZBFW uses a new configuration framework called CPL (Cisco Policy Language, based on MQC).
- CPL Configuration Steps:

1- Define zones - Decide on the interface groupings, e.g. inside, DMZ, outside, etc.
2- Create the ACLs - Matching specific traffic.
3- Define class-maps - Reference the matched traffic.
4- Define policy-maps - Execute the wanted actions.
5- Define zone-pairs - Direction of traffic flow.
6- Apply policy-maps to zone-pairs - Applies a unidirectional policy.
7- Assign interfaces to zones

- Typical memory usage:

> Each TCP or UDP (layer3/4) session takes approx. 600 bytes of memory.
> Different protocols or application channel sessions might use more than 600 bytes of memory.
>> E.g. voice uses two channels, one for voice and one for signaling.

- Typical performance counters (claimed at least by Cisco, so assume half the values):

PLATFORM	THROUGHPUT	MAX CONCURRENT CONNECTIONS	MAX CONNECTIONS PER SECOND
1861	90 Mbps	75000	710
2821	352 Mbps	94000	1500
2851	452 Mbps	98000	2000
3825	564 Mbps	146000	3800
3845	729 Mbps	176000	6700

```
CONFIG-SET: Zone-Based Policy IOS Firewall
+---------------------------------------------
|     access-list 199 permit 10.0.0.0 0.0.0.255 any
|     !
|     class-map type inspect match-all HTTP-TRAFFIC         - Creates the inspect class-map
|      match protocol http                                  - Matches HTTP traffic
|      match access-group 199                               - And traffic matching ACL-199
|     !
|     policy-map type inspect MY-POLICY                     - Layer3/4 top-level inspect policy
|      class type inspect HTTP-TRAFFIC                      - Calls the class-map
|       inspect                                             - Defines the action
|       police 512000 burst 16000                           - Defines the aggregate police rate
|     !
|     zone security OUT                                     - Creates and labels the security zones
|      description Internet-Side
|     zone security IN
|      description LAN-Side
|     !
|     zone-pair security ZONE-PAIR source IN destination OUT
|      service-policy type inspect MY-POLICY                - Assigns the inspect policy-map to the direction of traffic
|     !
|     int serial0/0
|      zone-member security OUT                             - Assigns the interfaces to zones
|     int ethernet0
|      zone-member security IN
```

```
----------
 COMMANDS
----------
# sh ip port-map                                - Shows a list of supported protocols available and the port-numbers
# sh policy-map type inspect zone-pair session  - Shows the stateful packet inspection sessions

#ip port-map {protocol} port {port} [acl]       - Add custom port-to-application mappings

#parameter-map type inspect {map-name}          >>> Configures an inspect parameter map <<<
 #alert {on | off}                              - Toggles packet inspection alert messages
 #audit-trail {on | off}                        - Turns audit trail messages on or off
 #tcp finwait-time {seconds}                    - Specifies how long a TCP session will be managed on FIN-exchange
 #tcp idle-time {seconds}                       - Configures the idle timeout for TCP sessions
 #tcp synwait-time {seconds}                    - Specifies how long IOS will wait for a TCP session to reach
                                                    established state before dropping the session
 #udp idle-time {seconds}                       - Configures the idle timeout for UDP sessions

#parameter-map type urlfilter {map-name}        >>> Creates a URL filtering parameter map <<<
  #server vendor websense {ip|hostname [port]}  - Specifies the URL filtering server
  #source-interface {interface}                 - Specifies source interface to be used when talking to the URL-server

#parameter-map type protocol-info {map-name}    >>> Defines an application-specific parameter map
  #server name {name}                           - Specifies the DNS name for MSN interaction
  #server ip {ip-add}                           - Specifies the IP of the server
```

```
#class-map type inspect [match-any|match-all] {name}  >>> Creates a layer 3 or layer4 inspect type class map <<<
 #match access-group {acl}                      - Use an ACL for matching
 #match protocol {protocol}                     - Reference a specific protocol signature
 #match class-map {class-name}                  - Reference another class-map for nesting

#policy-map type inspect {p-name}               >>> Creates a layer3 and layer4 inspect type policy map <<<
 #class type inspect {name}                     - Specifies the traffic (class) on which an action is to be performed
  #inspect [map-name]                           - Enables Cisco IOS stateful packet inspection
  #police rate {bps} burst {size}               - (o) Limits traffic matching within a firewall (inspect) policy
  #drop [log]                                   - (o) Drops matched packets within defined class
  #pass                                         - (o) Allows matched packets within defined class
  #service-policy type inspect {pair-name}      - Attaches a firewall policy map to a zone-pair
  #urlfilter {map-name}                         - (o) Enables Cisco IOS firewall URL filtering

#zone security zone-name                        - Creates a security zone
 #description {desc}                            - Describes the zone
#interface fa0/0
 #zone-member security {zone}                   - Assigns an interface to a specified security zone

#zone-pair security {zone-name} source {zone} destination {zone}
                                                - Creates a zone-pair
 #service-policy type inspect policy-map-name   - Attaches a firewall policy map to the destination zone-pair
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*======================================*
   IPS (Intrusion Prevention Systems)
*======================================*
- DOC-CD LOCATION
      > Cisco IOS Software Releases 12.4 T
       > Security and VPN
        > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4T
         > Configuring Cisco IOS Intrusion Prevention System (IPS)
```

- IPS helps to protect a network from both internal and external attacks and threats by making use of signatures.
- When loading signatures onto a router, either load the default, built-in signatures, or download the latest signatures from CCO via Security Device Manager (SDM), which also provides updates.

- The Cisco IOS IPS acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures.

- When packets in a session match a signature, the Cisco IOS IPS can take any of the following actions:

```
 > Send an alarm to a syslog server or a centralized management interface.
 > Drop the packet.
 > Reset the connection.
 > Deny traffic from the source IP address of the attacker for a specified amount of time.
 > Deny traffic on the connection on which the signature was seen for a specified amount of time.
```

- Individual signatures can be disabled in the case of false positives.

- An SDF (Signature Definition File) has definitions for each signature it contains.
- After signatures are loaded and compiled onto a router running Cisco IOS IPS, IPS can begin detecting the new signatures immediately.
- If the default, built-in signatures are not used, then one of three different types of SDF files can be selected for download, they are pre-configured for routers with memory requirements via the Flash memory:

```
 > attack-drop.sdf file
      >> For routers with less than 128MB memory, contains 80+ signatures.
 > 128MB.sdf
      >> For routers with more than 128MB memory, contains 300+ signatures.
 > 256MB.sdf
      >> For routers with more than 256MB memory, contains 500+ signatures.
```

- Cisco IOS IPS uses SMEs (Signature Micro Engines) to load the SDF and scan signatures.
- Signatures contained within the SDF are handled by a variety of SMEs.
- The SDF typically contains signature definitions for multiple engines.
- The SME typically corresponds to the protocol in which the signature occurs and looks for malicious activity in that protocol.
- A packet is processed by several SMEs. Each SME scans for various conditions that can lead to a signature pattern match.
- When an SME scans the packets it extracts certain values, searching for patterns within the packet via the regular expression engine.
- Refer to the DOC-CD for a list of supported signature engines.
- Refer to the DOC-CD for a list of alarm, status and error messages.

- Either the default, built-in signatures or an SDF example "attack-drop.sdf" may be loaded — but not both.
- If IPS cannot load the attack-drop.sdf file onto a router, by default the router will revert to the built-in signatures.

COMMANDS

```
# sh ip ips configuration                   - Shows the IPS configuration
# sh ip ips signatures [detailed]           - Shows signature configuration, including disabled signatures


#ip ips sdf location {URL}                  - (o) Specifies the location of the SPF to be loaded
                                            - If command is not issued, built-in signatures are loaded
#no ip ips location in builtin              - (o) Instructs the router to not load the built-in signatures if it
                                                  cannot find the specified .sdf signature file

#ip ips name {ips-name} [list acl]          - Creates an IPS rule
#ip ips signature {sign-id}  {delete | disable | acl}
                                            - (o) Attaches a policy to a given signature
#ip ips deny-action ips-interface           - (o) Creates an ACL filter for the deny actions on the IPS interface
                                                   rather than the ingress interface
#ip ips fail closed                         - (o) Drop all packets until the signature engine is built and ready

#interface fa0/2
 #ip ips {ips-name} {in | out} [list acl]   - Applies the IPS rule, loads the signatures and builds the engines
                                            - [list] Packets permitted as per ACL will be scanned by IPS
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==============================*
   Common Number Ranges
*==============================*
- DOC-CD LOCATION
       > Firewall Appliances
        > Cisco ASA 5500 Series Adaptive Security Appliances, Configuration Guides
           > Cisco ASA 5500 Series Configuration Guide using the CLI, 8.2
            > References :  Addresses, Protocols and Ports

- Protocol Numbers
      1     -     ICMP
      2     -     IGMPv1
      6     -     TCP
      17    -     UDP
      41    -     IPv6
      47    -     GRE
      50    -     ESP
      51    -     AH
      88    -     EIGRP
      89    -     OSPF
      103   -     PIM
      112   -     VRRP

- Port Numbers
      20          tcp   -     FTP data
      21          tcp   -     FTP control
      22          tcp   -     SSH
      23          tcp   -     Telnet
      25          tcp   -     SMTP
      53    udp         -     DNS query (this is used to translate www.google.com to an IP)
      53          tcp   -     DNS zone transfer
      67    udp         -     BOOTP Server
      68    udp         -     BOOTP Client
      69          tcp   -     TFTP
      80          tcp   -     HTTP
      123         tcp   -     NTP
      161   udp   tcp   -     SNMP
      162   udp   tcp   -     SNMP trap
      179         tcp   -     BGP
      443   udp   tcp   -     HTTPS
      445         tcp   -     MS-DS
      500   udp         -     ISAKMP
      520   udp         -     RIP
      1433  udp   tcp   -     MS-SQL Server
      1434  udp   tcp   -     MS-SQL Monitor
      1985  udp         -     HSRP
      2048  udp         -     WCCP
```

```
- Port Ranges
      Well-known Ports        -     0-1023
      Registered (high) Ports -     1024-49151
      Private Ports           -     49152-65535
      IP RTP (voice)          -     16384-32767

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==============================*
   Security Compliance RFCs
*==============================*
RFC 1918
---------
      10.0.0.0/8
      172.16.0.0/12
      192.168.0.0/16

RFC 3330 (More relevant to the SP track, but good to know for production networks) (RFC 5735 Obsoletes RFC3330)
---------
      0.0.0.0/8
      14.0.0.0/8
      24.0.0.0/8
      39.0.0.0/8
      127.0.0.0/8
      128.0.0.0/8
      169.254.0.0/16
      191.255.0.0/16
      192.0.0.0/24
      192.0.2.0/24
      192.88.99.0/24
      192.18.0.0/9
      223.255.255.0/24
      224.0.0.0/12
      240.0.0.0/12

RFC 2827
--------
      173.1.0.0/16


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================*
   TCP Intercept
*=====================*
- DOC-CD LOCATION
       > Cisco IOS Software Releases 12.4T Configuration Guides
        > Security and VPN
         > Security Configuration Guide: Securing the Data Plane, Release 12.4T
          > Configuring TCP Intercept
```

- A SYN flood DOS attack is when a source/s send a flood of thousands of TCP SYN packets usually containing a bogus source IP address. The receiving server would normally respond with a SYN/ACK and wait for the source to complete the handshake by sending an ACK. Because the ACK is not received, the session is kept open until it expires before it is torn down and the resources reallocated by the server. As a result, the server runs out of resources and is unable to establish legitimate TCP sessions.

- TCP Intercept can be used to help prevent TCP SYN flood DOS attacks, by allowing a router to intercept the initial SYN, and respond with a SYN/ACK. If the ACK was received, the session is forwarded on to the server, otherwise a RST is generated.

```
- Used to prevent TCP-SYN DOS attacks.
- Attacked would sent only SYN packets, but never completes the connection.
- Two modes:
      > Watch           - This mode just monitors the TCP setup and if half-open sessions are picked up, it will send the
                              SYN/ACK to the receiver.
      > Intercept       - This mode actually proxies the TCP setup and intercepts the TCP sessions.

- Optionally, an ACL can be used to restrict which hosts should be watched.
```

```
-----------
 COMMANDS
-----------
# sh tcp intercept statistics                   - Shows TCP intercept statistics
# sh tcp intercept connections                  - Shows incomplete connections and established connections

#ip tcp intercept list {acl}                    - Used to restrict which hosts are being watched
#ip tcp intercept watch-timeout {sec}           - Time to wait for a session to complete handshake
#ip tcp intercept mode {watch|intercept}        - Changes the mode (default = watch)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   IP Source Tracking
*=========================*
- DOC-CD LOCATION
       > Cisco IOS Software Releases 12.4T Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing User Services, Release 12.4T
          > IP Source Tracker
```

- The IP Source Tracker feature allows information to be gathered about the traffic flowing to a host that is suspected of being under attack.
- This feature also allows an attack to be easily traced to its entry point into the network.

```
-----------
 COMMANDS
-----------
# sh ip source-track summary                   - Shows traffic flow statistics

#ip source-track {IP}                          - Enables IP source tracking for a destination address
#ip source-track address-limit {ACL}           - (o) Limit hosts that can be simultaneously tracked at any given time
#ip source-track syslog-interval {minutes}     - (o) Sets the time interval, used to generate syslog messages (def=none)


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================*
   IP Traffic Export
*=======================*
- DOC-CD LOCATION
       > Cisco IOS Software Releases 12.4T Configuration Guides
        > Cisco IOS Security Configuration Guide: Securing User Services, Release 12.4T
         > User Security Configuration
          > IP Traffic Export

- Without the ability to export IP traffic, the Intrusion Detection System (IDS) probe must be in line with the network device
       to monitor traffic flow.
- IP traffic export eliminates the probe placement limitation, allowing users to place an IDS probe in any location within
      their network or to direct all exported traffic to a VLAN that is dedicated for network monitoring.
- Allowing users to choose the optimal location for their IDS probe reduces processing burdens.

-----------
 COMMANDS
-----------
# sh ip traffic-export {interface} {profile}   - Shows information related to exported IP traffic events

#ip traffic-export profile {name}              - Creates or edits an IP traffic export profile
 #interface fa0/0                              - Specifies the outgoing (monitored) interface for exported traffic
 #bidirectional                                - (o) Exports incoming and outgoing IP traffic on the interface
                                                     (default = inbound only)
 #mac-address {h.h.h}                          - (o) Specifies the 48-bit address of the destination host
 #incoming access-list {acl}                   - (o) Configures filtering for incoming traffic
 #outgoing access-list {acl}                   - (o) Configures filtering for outgoing export traffic
 #exit
#inteface fa2/1
 #ip traffic-export apply {name}               - Enables IP traffic export on an ingress interface

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================*
    URPF
*====================*
- DOC-CD LOCATION
       > Cisco IOS Software Releases 12.4T Configuration Guides
        > Security Configuration Guide: Securing the Data Plane, Release 12.4T
         > Configuring Unicast Reverse Path Forwarding
```

- When Unicast RPF is enabled on an interface, the router examines all packets received as input on that interface to ensure that the source address and source interface appear in the routing table and match the interface on which the packet was received.
- This 'look backwards' ability is available only when CEF is enabled on the router, because the lookup relies on the presence of the Forwarding Information Base (FIB). CEF generates the FIB as part of its operation.
- URPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.
- URPF must be applied at the interface downstream from the larger portion of the network, preferably at the edges of the network.

- If the packet was received from one of the best reverse path routes, the packet is forwarded as normal.
- If URPF does not find a reverse path for the packet, the packet is dropped or forwarded, depending on whether an ACL is specified
- URPF considers all equal-cost 'best' return paths to be considered valid. This means that URPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost and as long as the route is in the FIB.

- There are two modes:

 > Strict URPF mode

>>A Strict mode check is successful when URFP finds a match in the FIB for the packet source address and the ingress interface through which the packet was received matches one of the URPF interfaces in the FIB match. If this check fails, the packet is discarded. This type of URPF check can be used where packet flows are expected to be symmetrical.

> Loose URPF mode

>>A Loose mode check is successful when a lookup of a packet source address in the FIB returns a match and the FIB result indicates that the source is reachable through at least one real interface. The ingress interface through which the packet was received is not required to match any of the interfaces in the FIB result.

```
CONFIG-SET: URPF - Log every 10th Denied Spoofed Packet
+----------------------------------------------------------
|     access-list 100 deny ip any any log                 - Creates the ACL-100 to log denied traffic
|     access-list log-update threshold 10                 - Sets ACLs to log every 10th entry
|     !
|     interface Serial 0/0
|      ip verify unicast source reachable-via rx 100      - Enables URPF on the interface referencing ACL-100
|
```

```
-----------
 COMMANDS
-----------
#ip cef                                           - Enables CEF, this is required
#interface fa0/0
 #ip verify unicast reverse-path [acl]            - Enables Unicast RPF on the interface (LEGACY COMMAND)
                                                  - [ACL] Permits: spoofed packets are permitted
                                                  - [ACL] Denies: spoofed packets are dropped

 #ip verify unicast source reachable-via {any [allow-default] | rx}
                                                  - Configures Unicast RPF on the interface
                                                  - [any] Specifies loose Unicast RPF
                                                  - [rx] Specifies strict Unicast RPF
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================================*
    Local Authentication & Privilege
*=====================================*
- DOC-CD LOCATION
       > Cisco IOS Software Releases 12.4T Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing User Services, Release 12.4T
          > Configuring Security with Passwords, Privilege Levels and Login Usernames

- Before securing a device, it should be understood that the Cisco IOS command-line interface is divided into different
      command modes. Here are some well-known modes:
 > User EXEC Mode
      >> User exec mode is set by default to privilege level 1, which is the first level when logged into a router.
      >> This mode provides limited access to exec commands (exec commands being the show and clear commands)
      >> Secure this mode by setting terminal line passwords, i.e. vty, console and aux.
      >> Default prompt for this mode is "Router>".

 > Privileged EXEC Mode
      >> Also known as Enable mode.
      >> In order to have access to all exec commands, a privileged-level password must be entered.
      >> Once in Privileged exec mode, any EXEC command can be entered.
      >> Privileged exec mode is set by default to privilege level 15.
      >> "enable" and "disable" commands are used to navigate to and from Privileged exec mode.
      >> Secure this mode with the "enable password" or "enable secret".
      >> Default prompt for this mode is "Router#"

 > Global Configuration Mode
      >> Global configuration mode is used to configure the system globally, or to enter specific configuration modes.
      >> Default prompt for this mode is "Router(config)#".
      >> The default privilege level is 15 for users.
      >> Command used to enter is "config terminal" and "exit" or Ctrl-Z to leave.
      >> Secure this mode by defining the privilege levels and assigning command and user accounts to the different levels.

- The 'privilege' command is used to move commands from one privilege level to another in order to create  additional levels
      of administration of a networking device.
- This is required by companies that have different levels of network support staff with different skill levels.

CONFIG-SET: Privilege-levels to only allow certain fields in a "SHOW RUN" for privilege-level-2 users
+------------------------------------------------------
|     username users privilege 2 password Limit3d     - Creates the users account to only see privilege level-2
|     !                                                     when logged in
|     privilege configure level 2 hostname            - Allows output to list the router hostname
|     privilege configure level 2 interface           - Allows output to list interfaces
|     privilege interface level 2 ip access-group     - Allows output to list ACLs applied to interfaces
|     privilege interface level 2 encapsulation       - Allows output to list of encapsulations
|     !
|     privilege exec level 2 show running-config      - Specifies the command allowed to be executed
|
----------
 COMMANDS
----------
# sh privilege                                    - Will display the current privilege level
# enable 15                                       - Will allow a user to enter a higher privilege level

#service password-encryption                      - Enables password encryption for all passwords clear text passwords
#enable secret {PWD}                              - Sets a privilege exec encrypted password
#username Tea-Tady privilege 1 password 2SUGARS - Setup a user to have privilege level 1 when logging into the router
#username Norman privilege 2 password Limit3d    - Setup a user to only see privilege level 2 when logged
#username Geek privilege 15 password 1337        - Setup a user to login with full privileges

#privilege exec [all] level {level} {command-string}
                                                  - Assigns commands to specific privilege levels
                                                  - [all] All sub-options will be set to the same level

#privilege {configure|interface...} {level} {string}
                                                  - Specify what is allowed in the output sections

#line vty 0
 #login                                           - Use the password specified next for VTY access on line 0
 #password {PWD}                                  - Sets the user exec level password for VTY terminal access

#line vty 1-2
 #login local                                     - VTY access on line 1-2 will the local username database



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=======================================================*
   AAA (Authentication, Authorization, Accounting)
*=======================================================*
- DOC-CD LOCATION
       > Cisco IOS Software Releases 12.4T Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing User Services, Release 12.4T
          > Authentication, Authorization, Accounting
           > Copy descretly owned by Kane  Bagwell

- Full AAA knowledge is outside of the scope of the R&S lab exam (only the basic IOS config should be known).
- Authentication provides the method of identifying users, including login and password dialog and possibly encryption.
- Authentication is the way a user is identified prior to being allowed access to the network and its services.

- AAA Authentication Login Methods:
 > Enable                 - Uses the enable password for authentication.
 > Line                   - Uses the terminal line password for authentication.
 > Local                  - Uses the local username database for authentication.
 > Local-Case             - Uses Case-Sensitive local username authentication.
 > None                   - Uses no authentication.
 > Group radius           - Uses the list of all RADIUS servers for authentication.
 > Group tacacs+         - Uses the list of all TACACS+ servers for authentication.

- The AAA authorization feature is used to determine what a user may and may not do.
- When AAA authorization is enabled, the user is granted access to a requested service only if the user is allowed it.

- AAA Authorization Types (of relevance to R&S):
 > Exec                  - Applies to the attributes associated with a user exec terminal session.
 > Command               - Applies to the Exec mode commands a user issues. Command authorization attempts authorization
                               for all Exec mode commands associated with a specific privilege level.

- AAA supports Five different Methods of Authorization:
 > Tacacs+               - TACACS server is queried for authorization.
 > Radius                - RADIUS server is queried or authorization
 > If-authenticated      - The user is allowed to access the requested function provided the user has
                               been authenticated successfully.
 > None                  - The network access server does not request authorization information.
 > Local                 - The router consults its local database, as defined by the 'username' command. Only a limited set
                               of functions can be controlled through the local database.


----------
 COMMANDS
----------
#aaa new-model                                 - Enables AAA globally

#aaa authentication login {default | listname} method1 [method2...]
                                               - Configures authentication lists for logins to the device

#aaa authentication password-prompt C:\        -(o) Changes the text displayed when a user is prompted for password
#aaa authentication banner @ WELCOME SIR @     -(o) Creates a personalized login banner
#aaa authentication fail-message @ HAHA @      -(o) Creates a message to be displayed when a user fails login

#aaa authorization {exec|commands} {default | list-name} method1 [method2...]
                                               - Configure authorization to determine device access

#no aaa authorization config-commands          - (o) Disables authorization for all global configuration commands

#line vty 0 4
 #login authentication {listname}              - VTY access will use AAA to query local user database
 #timeout login response {sec}                 - (o) How long the system will wait for login information before timing out
 #authorization {exec|commands} {level} {name}  - Applies the authorization list to a line or set of lines
```

```
  ______                  _
 / _____)                (_)
( (____  _____  ____ _   _ _  ____ _____  ___
 \____ \| ___ |/ ___) | | | |/ ___) ___ |/___)
 _____) ) ____| |    \ V /| ( (___| ____|___ |
(______/|_____)_|     \_/ |_|\____)_____|___/
```

```
*-=-=-=-=-=-=-=-=-=-=-=-*
|      INDEX            |
*-=-=-=-=-=-=-=-=-=-=-=-*
- DHCP (Dynamic Host Configuration Protocol)
      + DHCP Clients
      + DHCP Server
            o Manual Bindings
            o DHCP Server Options
- DNS (Domain Name System)
      + Static DNS Entries
      + DNS Clients
      + DNS Proxy
      + DNS Server
      + Authoritative DNS Server
- MTU (Maximum Transmission Unit)
      + TCP MSS (Maximum Segment Size)
      + PMTU Discovery (Path MTU)
- ICMP (Internet Control Message Protocol)
      + Traceroute
      + ICMP, UDP and TCP Implementations
      + Ping
      + ICMP Rate-Limit
- IRDP (ICMP Router Discovery Protocol)
- IP SLA and Object Tracking
- First Hop Redundancy
      + HSRP (Hot Standby Router Protocol)
      + VRRP (Virtual Router Redundancy Protocol)
      + GLBP (Gateway Load Balancing Protocol)
- NAT (Network Address Translation)
      + Static NAT
      + NAT-Pool
      + Overload
      + NAT timeouts
      + NAT TCP Load Balancing
      + NAT-on-a-Stick
      + NAT Order of Operation
- NTP (Network Time Protocol)
      + Polling Modes
            o Client/Server Mode
            o Peer Mode
      + Broadcast Mode
      + Multicast Mode
```

- SNMP (Simple Network Management Protocol)
  + Message types
  + IfIndex
  + Polling & Trapping
  + SNMP Community String
- RMON (Remote Monitoring)
  + Delta Sampling
  + Absolute Sampling
- Syslog
  + Logging Levels
  + Logging History
  + Config Change Notification and Logging (Archive)
- Netflow
  + Flows
  + Top-Talkers
- RITE (Router IP Traffic Export)
- IP Accounting
- VTY Access Using Telnet
  + Login Enhancements
  + Rotary Group
  + Various telnet Commands
  + VTY Timeouts
- VTY Access using SSH (Secure Shell)
- SCP (Secure Copy)
- Banners
  + Banner Types
  + Banners Using Tokens
- IOS Menus
- HTTP Server
- TFTP Server
- FTP Server
- CDP (Cisco Discovery Protocol)
- WCCP (Web Caching Content Protocol)
- IP and Command Aliases
- IP Event Dampening
- Crash Dump
- Warm Reload
- System Resources
  + CPU Threshold Notification
  + Memory Threshold Notification
- KRON Command Scheduler
- EEM (Embedded Resource Manager)
- Other Services
  + TCP Extensions for High Performance
    - TCP Selective Acknowledgment
    - TCP Time-Stamp
    - TCP Window Scaling
    - TCP ECN
  + TCP Keepalive
  + TCP Synwait-Time

```
      + TCP/UDP Small Services
      + Service Nagle
      + Scheduler Allocate
      + DRP Server Agent
- Disabling Unnecessary Services
      + Source Routing
      + BOOTP and DHCP
      + CDP
      + Finger
      + IP ICMP Redirect
      + Proxy ARP
      + IP-Unreachables
      + IP Mask-Reply
      + IP Directed Broadcast
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==================================================*
    DHCP (Dynamic Host Configuration Protocol)
*==================================================*
- DHCP stands for Don't Hit Computer People!
- DOC-CD LOCATION
       > 12.4T Configuration Guides
         > Cisco IOS IP Addressing Configuration Guide, Release 12.4T
          > Part - 3 DHCP


- DHCP Clients: IP Helper and DHCP Relay
 > A router is a BOOTP server by default.
 > Bootp requests can be forwarded by using the "ip helper-address" command.

- Gothas to look out for!
 > Be sure to check for excluded IPs already in use from the pool, like the HSRP address, interface addresses, gateway, dns, etc.
 > The pool should consist of all valid host IP's in the lower /25.
      >> Pool range = 129 - 254, and NOT 128 - 255.
 > If DHCP snooping was configured as part of the DHCP switching config, then you must enable the port
      connecting to the DHCP server as a trusted port.

- Frame-Relay Client with DHCP
 > Pre-configuring Frame-Relay clients requesting a DHCP address can be done using:
      #frame-relay interface-dlci 555 protocol ip 192.168.1.5

- DHCP Server
 > Configuring
       1 - Configure the router to exclude its own IP address, and other necessary IPs from the DHCP pool.
       2 - Configure the DHCP pool, gateway, name-servers, and other options.
       3 - Disable DHCP conflict logging or configure a DHCP database agent.
 > Configuring Manual Bindings
      >> All DHCP clients send a client identifier (DHCP option 61) in the DHCP packet.
      >> Used to force a client to get the same DHCP IP based on its MAC-address.
      >> 01 is prepended to the MAC used in the 'client-identifier'.
      >> Example:
         >> Client NIC has a MAC: 001d-0948-9857.
         >> Add 01 to the front: 01001d-0948-9857.
         >> Convert to IOS MAC format: 0100.1d09.4898.57.

 > DHCP Server Options
      >> Option 12 - Specifies the hostname of the client.
      >> Option 51 - Allows the client to request a lease time for the IP address.
      >> Option 55 - Allows the DHCP client to request certain options from the DHCP server.
      >> Option 60 - Allows the user to configure the vendor class identifier string to use in the DHCP interaction.
      >> Option 61 - This option is used by DHCP clients to specify their unique identifier, typically the MAC.
      >> Option 66 - Hand-out IP address of TFTP server.
      >> Option 82 - DHCP-Relay.
```

```
CONFIG-SET: DHCP server configuration
+---------------------------------------------------------
|     ip dhcp excluded-address 150.100.1.101                - Excludes one IP from the DHCP pool
|      ip dhcp database flash:/bindings                     - Stores the DHCP bindings in flash memory
|     !
|     ip dhcp pool DHCP
|      network 150.100.1.0 255.255.255.0
|      bootfile R7-config                                   - Specifies a config file the client will load
|      option 150 ip 150.100.3.59                           - Same as next-server command
|      default-router 150.100.1.4                           - Specifies the default gateway
|      lease 0 20                                           - Configures lease for 20 hours
|
```

```
----------
 COMMANDS
----------
# renew dhcp {int}                             - Requests new IP via DHCP for the interface
# release dhcp {int}                           - Release the DHCP IP for the interface
# sh ip dhcp database                          - Shows database settings
# sh ip dhcp database bindings                 - Shows the bindings

# debug dhcp detail                            - Great debug command to see most DHCP information
# debug ip dhcp server packets                 - Shows packet level detail
# debug ip dhcp server events                  - Shows DHCP events and negotiations

#frame-relay interface-dlci {dlci} prot ip {IP} - Allows pre-configuring a new frame relay neighbor
                                               - {IP} Will be assigned to the neighbor using DHCP
#no ip bootp server                            - Disables the BOOTP service, (enabled by default)
#ip dhcp excluded-address low-address [high-address]
                                               - Specifies the IP addresses to be excluded from the DHCP pool
```

```
#ip dhcp relay info {check|option|trust}        - Specifies DHCP relay agent properties
                                                 - {check}: Validate relay information in BOOTREPLY
                                                 - {option}: Insert relay information in BOOTREQUEST
                                                 - {trust}: Received DHCP packets may contain relay info option

#ip dhcp database [url:/name]                    - Specifies a location to store DHCP bindings
#no ip dhcp conflict-logging                     - Disables DHCP conflict logging
#ip dhcp pool {name}                             - Creates a DHCP Server Pool, and enters the DHCP-config-mode
 #network {subnet} {mask}                        - Specifies the subnet network number and mask of the DHCP address pool

 #domain-name {domain-name}                      - Specifies the domain name for the clients
 #dns-server {ip} [ip2 ip3..]                    - Specifies the IPs of a DNS server to a DHCP client
 #default-router {ip} [ip2]                      - (o) Specifies the IP address of the default router/s
                                                 - The IP address should be on the same subnet as the client
 #lease {days [hours] [minutes] | infinite}      - (o) Specifies the duration of the lease  (default = 1 day)

#interface fa0/1                                 >>> DHCP CLIENT CONFIG <<<
 #ip address dhcp[client-id fa0/1]               - Configure the interface to request DHCP IP
 #ip dhcp client hostname ROUTER3                - Sets option 12, the hostname
 #ip dhcp client lease {days hours min}          - Sets option 55, lease timers

#interface fa0/2                                 >>> DHCP RELAY CONFIG <<<
 #ip helper-address {ip}                         - Relay BOOTP requests to a DHCP server

#interface fa0/3
 #ip dhcp relay information trusted              - Enables forwarding DHCP requests containing option 82 info


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==================================*
    DNS (Domain Name System)
*==================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
         > Cisco IOS IP Addressing Configuration Guide, Release 12.4T
          > Part - 4 DNS

- Static DNS Entries (name-to-IP):
            #ip host TheNewGuy 10.1.1.1                    - Creates a static mapping

- DNS Client Config
            #ip domain-lookup                              - Enables DNS lookups for queries (enabled by default)
            #ip name-server 172.60.60.1 172.80.80.1        - Specifies the DNS servers to query
            #ip domain name bob.com                        - (o) Specifies the  local domain

- DNS Proxy
            #ip dns server - Enable DNS server
            #ip dns spoofing                               - Enables spoofing replies to DNS queries
```

- Simple DNS Server Config

```
            #ip dns server                          - Enables the DNS server
            #ip domain-lookup                       - Enables DNS lookups for queries (enabled by default)
            #ip name-server 146.6.6.1 148.8.8.1     - Specifies the DNS servers to query
```

- Authoritative DNS Server Config
  > Using your router as a DNS server is not recommended, but it is possible.

```
      Step 1- Enable the enable DNS server:
            #ip dns server

      Step 2- Create the primary DNS record and optionally the DNS refresh timers:
            #ip dns primary website.com soa ns.website.com admin@website.com 86400 3600 1209600 86400

      Step 3- Define primary and secondary name servers for the domain:
            #ip host website.com ns ns.website.com
            #ip host website.com ns ns.isp.com

      Step 4- Define mail records for the domain with the "ip host mx" command:
            #ip host website.com mx 10 mail.website.com
            #ip host website.com mx 20 mail.isp.com

      Step 5- Finally, you need to define hosts within your domain:
            #ip host ns.website.com 192.168.0.1       - Router's IP address
            !
            #ip host www.website.com 192.168.1.1
            #ip host website.com 192.168.1.1          - Alternate for www.website.com
            !
            #ip host mail.website.com 192.168.1.2
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
```

MTU (Maximum Transmission Unit)

```
*==================================*
```

- DOC-CD LOCATION
  > 12.4T Configuration Guides
    > Cisco IOS IP Application Services Configuration Guide, Release 12.4T
      > Configuring TCP

- MTU
  > The size (in bytes) of the largest PDU (Protocol Data Unit) that an interface can pass onwards without the need to fragment.
  > All interfaces have a default MTU packet size.
  > The IP MTU size can be adjusted so that the Cisco IOS software will fragment any IP packet that exceeds the MTU set for the interface.
  > Changing the MTU value can affect the IP MTU value. If the current IP MTU value is the same as the MTU value and the MTU value is changed, the IP MTU value will be modified automatically to match the new MTU. However, the reverse is not true; changing the IP MTU value has no effect on the value for the MTU interface configuration command.

```
- TCP MSS (Maximum Segment Size)
 > Enables the configuration of the MSS for transient packets that traverse a router, specifically TCP
     segments in the SYN bit set.
 > When a host/PC initiates a TCP session with a server, by using the MSS option field in the TCP SYN packet, the maximum payload
     size is negotiated to make sure no fragmentation would be needed once the data is sent.
 > The MSS is governed or determined by the MTU of the link.
 > This payload size excludes the transmit overhead. For example, the following:
       - IP header (20-byte).
       - TCP header (20-byte).
       - PPPoE header (8-byte).
 > In most cases, the optimum MSS value is 1460 bytes. This value plus the 20-byte IP header, and the 20-byte TCP header,
     add up to a 1500-byte packet that matches the MTU size for the ethernet link.

- PMTU Discovery (Path MTU)
 > Method for maximizing the use of available bandwidth in the network between the endpoints of a TCP connection.
 > IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable MTU size of
     the various links along the path
 > IPv4 default - TCP Path MTU Discovery is disabled. If enabled, the default is 10 minutes.
 > IPv6 default - TCP Path MTU Discovery is enabled.
 > All TCP sessions are bound by a limit on the number of bytes that a single packet can transport.
     This limit, known as the MSS, is 536 bytes by default.
 > In other words, TCP breaks up packets in a transmit queue into 536-byte chucks, before passing them down to the IP layer.
 > PMTU can be enabled to dynamically determine how large the MSS can be without creating that needed fragmentation.
 > TCP then uses this MTU value, minus room for IP and TCP headers, as the MSS for the session.
 > This feature is described in RFC 1191.
```

```
 ----------
 COMMANDS
 ----------
# sh ip bgp neighbors | i max data                 - Shows the MSS for BGP neighbors

#interface fa0
 #mtu{size}                                        - Sets the interface MTU for all protocols, in bytes
 #ip mtu {size}                                    - Sets the IP MTU size of IP packets, in bytes, sent on an interface

 #ip tcp adjust-mss {size}                         - Adjusts the MSS value of TCP SYN packets going through a router
                                                   - {size} specified in bytes
                                                   - The range is from 500 to 1460

#ip tcp mss {size}                                 - Changes (MSS) for TCP connections originating or terminating on a router
                                                   - Disables by default. If this command is not enabled, the MSS value
                                                   of 536 bytes is used if the destination is not on a LAN,
                                                   otherwise the MSS value is 1460 for a local destination

#ip tcp path-mtu-discovery [age-timer {minutes | infinite}]
                                                   - Enables the PMTU discovery feature for all new TCP connections
                                                   - The age timer how often TCP re-estimates the path MTU with a larger MSS
                                                   - (default = 10min)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==============================================*
   ICMP (Internet Control Message Protocol)
*==============================================*
- Two steps are involved with a Traceroute:
 1- Manipulating the TTL in the IP header to find the routers in the path to the destination.
      >> The source initiating the trace will generate three ICMP echos towards the destination, starting with a TTL=1.
      >> Each router in the path decrements the IP TTL by 1 upon receipt of the packet.
      >> If a router in the path decrements the received packet's TTL to 0, it will discard the packet and
            generate an ICMP 'time-exceeded' message to indicate to the source that the packet expired in transit.
      >> Every time the source gets a 'time-exceeded' it will generate three new echos with the previous TTL incremented by 1.
      >> This cycle continues until the router receiving the ICMP packets matches the destination IP specified to one of its own.
 2- Getting some form of response from the destination to know if it was reached.
      >> After the destination is reached, the reply will depend on the packet type used by the traceroute application.
      >> If UDP was used, the packets sent to the destination would be sent to incremented unused UDP Ports. When
            the final destination receives these packets sent to an unused local UDP port, it will respond with an ICMP
            port-unreachable message. Once the source receives the ICMP port-unreachable, it knows the destination
            was reached.
      >> If ICMP was used, the process is the same as before, but the destination will reply with an ICMP echo-reply.

- Three different implementations of traceroute:
 > ICMP
      >> Used natively by Windows. Also supported by Linux.
 > UDP
      >> Used natively by Cisco routers starting at UDP port 33434.
      >> Used natively by Linux.
 > TCP
      >> Possible via 3rd party applications, and on Linux


- Allowing ping and traceroute traffic in ACLs:
 > Outbound Traffic
      >> ICMP echo                - Used by ping.
      >> ICMP echo                - Also used by ICMP-based traceroute applications.
 > Inbound Reply Traffic
      >> ICMP time-exceeded       - Needed for the replies from the routers in the transit paths.
      >> ICMP port-unreachable    - Needed to indicate that the destination was reached, if a UDP-based application was used.
      >> ICMP echo-reply          - Needed to indicate that the destination was reached, if an ICMP-based application was used.
                                  - Would be dynamically included in a reflexive ACL.

- Ping
 > Ping is NOT an acronym, many believe 'ping' is short for Packet INternet Groper, but that is not the case.
 > The author Mike Muuss, named 'ping' after the sounds a sonar makes, because of operational similarities.
 > The Cisco "ping" command sends an echo request packet to an address, then waits for a reply.
 > Ping output can help evaluate the following:
      >> Path-to-host reliability.
      >> Delays over the path.
      >> Whether the host can be reached, or is functioning.
 > Ping extended mode is invoked by just entering "ping" without any options.
 > The output character of ping:
   >> !            - Each exclamation point indicates receipt of a reply.
   >> .            - Each period indicates that the network server timed out while waiting for a reply.
   >> U            - A destination unreachable error PDU (Protocol Data Unit) was received.
   >> C            - A congestion experienced packet was received.
   >> I            - User interrupted test.
   >> M            - A destination unreachable error PDU was received.

- ICMP Rate-Limit
 > A built-in feature to protect a router against spoofed ICMP DOS attacks, by rate-limiting the amount of ICMP
      responses out of an interface for ICMP type-3 (port unreachable) and type-4 (fragmentation needed).
 > The effect of ICMP rate-limiting is typically seen as asterisk '*' on the last hop of a trace:
      Tracing the route to 192.168.10.1:
      1 192.168.7.5  7 ms  7 ms  5 ms
      2 192.168.10.1  16 ms *  16 ms

----------
 COMMANDS
----------
# traceroute [prot] [dst-ip] [source] [numeric] - To analyze the path to a destination
# ping [prot] [ip] [df|size|src|timeout|repeat] - To diagnose basic network connectivity

# show ip icmp rate-limit                       - Shows all current ICMP unreachable statistics

#ip icmp rate-limit unreachable [df] {rate}     - Changes how many unreachable the router will answer
                                                - [df] Optional, needed to also limit type-4 messages
                                                - {rate} Generate 1 response messages every 'x' ms
                                                - (default = One reply every 500 ms)


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================================*
   IRDP (ICMP Router Discovery Protocol)
*==========================================*
- DOC-CD LOCATION
        > 12.4T Configuration Guides
         > Cisco IOS IP Application Services Configuration Guide, Release 12.4T
          > Configuring First Hop Redundancy: IRDP

- IRDP allows hosts to locate routers that can be used as a gateway to reach IP-based devices on other networks.
- When the device running IRDP operates as a router, router discovery packets are generated.
- When the device running IRDP operates as a host, router discovery packets are received.

-----------
 COMMANDS
-----------
#interface fa0/1
 #ip irdp                                       - Enables IRDP on the interface
 #ip irdp holdtime {sec}                        - (o) Sets the IRDP period for which advertisements are valid
 #ip irdp maxadvertinterval {sec}               - (o) Sets the IRDP maximum interval between advertisements
 #ip irdp minadvertinterval {sec}               - (o) Sets the IRDP minimum interval between advertisements
 #ip irdp preference {number}                   - (o) Sets the IRDP preference level of the device
 #ip irdp address {IP} {number}                 - (o) Specifies an IRDP address and preference to proxy-advertise


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*================================*
   IP SLA and Object Tracking
*================================*
- IP SLA DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Network Management
         > Cisco IOS IP SLAs Configuration Guide, Release 12.4T

- TRACK DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Application Services Configuration Guide, Release 12.4T
         > Configuring Enhanced Object Tracking

- IP SLA was previously also known as
 > RTR (Response Time Reporter)
 > SAA (Services Assurance Agents)

- Trackable parameters:
 > Delay (UDP/VoIP).
 > Application response times (HTTP/DHCP/DNS/FTP).
 > Reachability (ICMP echo/UDP Echo/TCP Connect).

- Enhanced Object Tracking extends basic tracking to:
 > Interface line protocol status.
 > IP address lost (DHCP/IPCP).
 > Routing reachability.
 > Routing metrics.

- Refer to the following article on advance Track and IP SLA usage
      http://routing-bits.com/2009/07/24/using-the-track-statement/

-----------
 COMMANDS
-----------
# sh ip sla statistics
# sh ip sla monitor operational-state

#ip sla monitor 1                               - Create a SLA monitor
 #type pathEcho prot ipIcmpEcho {IP} src {IP}   - Create a ICMP-type SLA
 #frequency {seconds}                           - Specifies the frequency of the monitor
 #timeout {milliseconds}                        - How long to wait for an ICMP echo to timeout
 #request-data-size {bps}                       - Specifies the size of the echo's
 #threshold {ms}                                - Operation threshold in milliseconds
```

```
#ip sla monitor schedule 1 start-time {now|time} life {sec|forever}
                                              - Configures the scheduler to start now and continue running for so long

#track 1 interface serial0/0 line-protocol      - Track 1: The line protocol of serial 0/0
#track 2 ip route 192.168.0.0/24 metric thresh  - Track 2: The route 192.168.0.0/24 metric in the routing table
#track 3 ip route 192.168.1.0/24 reachability   - Track 3: The route 192.168.1.0/24 being in the routing table or not
#track 4 rtr 1 [reachability | state]           - Track 4: Use the IP SLA/RTR state/reachability to track. (RTR = IP SLA)
#track 5 list boolean {or|and}                  - Track 5: Make use of boolean and/or expression to groups objects together



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   First Hop Redundancy
*=========================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Application Services Configuration Guide, Release 12.4T
         > Configuring First Hop Redundancy


- HSRP (Hot Standby Router Protocol)
 > Cisco proprietary.
 > One ACTIVE router replies to ARP requests sent to a virtual IP address.
 > Uses UDP port 1985 for transport to destination 224.0.0.2.
 > Interface level commands "standby".
 > MAC: 0000.0c07.acxx - where XX is the HSRP group number in hex
 > Hello time = 3 sec.
 > Hold time = 10 sec.
 > Can use MSEC with HSRP v2.
 > Highest priority preferred, default = 100.
 > Roles can be changed only if preemption is enabled (disabled by default).
 > Authentication - plain text and MD5.
      >> A plain text password of 'cisco' is the default, and won't show up in the running config.
 > Router tracking uses a default decrement of 10.


- VRRP (Virtual Router Redundancy Protocol)
 > Open standard.
 > One MASTER router replies to ARP requests sent to a virtual IP address.
 > Transport protocol 112, multicast destination address 224.0.0.18.
 > Interface level command "vrrp".
 > VRRP work almost identically to HSRP with one big exception, with VRRP preemption is enabled by default.
 > VRRP master = HSRP active.
 > Pre-emption enabled by default.
 > Master 82 advertisement interval = 1 sec default.
 > Master down interval = 3.609 sec default.
 > MAC 0000.5e00.01xx - where XX is the VRRP group number in hex.
```

```
- GLBP (Gateway Load Balancing Protocol)
 > Cisco proprietary.
 > Two roles:
      >> Active virtual gateway.
      >> Active virtual forwarder.
 > Supports preemption.
 > Load-balancing algorithms:
      >> Round-Robin.
      >> Host dependent.
      >> Weighted.
 > Object tracking with GLBP based on weighting.

--------------
 COMMANDS
--------------
# sh standby                                 - Shows HSRP statistics, priority, counters, active and standby router
# sh vrrp                                    - VRRP config is virtually the same as HSRP, except for pre-emption
# sh glbp                                    - Show the GLBP statistics, priority, counters, etc

#track 1 rtr 1 state                         - Creates a track using SLA 1
 #delay up 10                                - If SLA is up for at least 10 sec, then the track kicks in

#interface fa0/0                             >>> HSRP-CONFIG <<<
 #standby [group] ip {virtual-ip}            - Defines the virtual IP to be used as a gateway
                                             - Group number determines the virtual MAC address
 #standby [group] timers {hello} {hold-time} - Changes the hello and hold time
 #standby [group] priority {1-255}           - Changes the priority. Higher preferred, (default = 100)
 #standby [group] preempt                    - Enables preemption
 #standby {group} mac-address                - Specifies a MAC to be used instead of the default (0000.0c07.acxx)
 #standby {group} use-bia                    - Use the interface-MAC appose to the HSRP MAC. Useful with "sw port-security"
 #standby {group} authentication             - Specifies authentication for the group
 #standby delay                              - Used to give IGP time to converge before previously active router comes up
 #standby {group} track {object} decrement {no} - If track even is successful, decrement the priority with configured value

#interface fa0/1                             >>> VRRP-CONFIG <<<
 #vrrp {group} ip {IP}                       - Defines the master IP to be used as a gateway
 #vrrp {group} timers advertise msec {msec}  - Changes the timers
 #vrrp {group} priority {1-255}              - Changes the priority. Higher preferred (default = 100
 #vrrp {group} authentication                - Specifies authentication for the group

#interface fa0/2                             >>> GLBP-CONFIG <<<
 #glbp {group} ip {virtual-ip}               - Defines the virtual IP to be used as a gateway
 #glbp {group} timers hellotime {holdtime}   - Changes the default timers
 #glbp {group} timers redirect {timeout}     - Changes the default redirect timers
 #glbp {group} load-balancing {round|host|weighted}
                                             - Specifies which algorithm to use
 #glbp {group} priority {level}              - Changes the priority
 #glbp {group} preempt [delay minimum {sec}] - Enables preemption
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================================*
   NAT (Network Address Translation)
*=======================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
        > Cisco IOS IP Addressing Services Configuration Guide, Release 12.4T
         > Part6: NAT

- Cisco NAT Whitepaper : http://tinyurl.com/nat-whitepaper
- NAT allows a host that does not have a valid registered IP address to communicate with other hosts on the Internet.
- NAT translates, or changes, one or both IP addresses (source and destination) inside a packet as it passes through a router.
- NAT Terminology:
 > IL (Inside Local)     - The local IP address of the private host on a network. Typically from private address space (RFC-1918).
 > IG (Inside Global)    - The public, registered IP address that the outside network sees as the IP address of your local host.
 > OL (Outside Local)    - The destination IP address, which the local host sees as the IP address of the remote host.
 > OG (Outside Global)   - The public, legal, registered IP address of the remote host (e.g. the IP address of the remote
                           web server that a PC is connecting to).

- Be careful when enabling NAT, to not NAT everything, otherwise locally-generated traffic, such as routing protocol traffic, will
     also get NATed. If the routing protocol traffic comes from an unknown source the routing protocols will break.

- Static NAT
 > A particular IL address always maps to the same IG (public) IP address.
 > If used, each Outside Local address always maps to the same Outside Global (public) IP address.
 > Example, internal host 192.168.10.1 will always be seen on the internet as 141.69.232.209.
     #ip nat inside source static 192.168.10.1 141.69.232.209 extendable
 > Instead of NATing whole IPs, NAT could also be used to on individual TCP/UDP ports, also known as NAT port-redirection.
 > Example, traffic on port-25 from host 192.168.10.1 will always be seen on the internet as 141.69.232.209 coming from port-2525
     #ip nat inside source static tcp 192.168.10.1 25 141.69.232.209 2525

- Inside Source
 > Most common implementation of nat. Used to hide private subnets (RFC-1918) behind one or more public IPs.
 > The words "inside source" emphasize that the inside source address is what's getting changed.
 > NAT-POOL Implementation
     >> Many Inside Local addresses are mapped to a POOL of Inside Global (public) IP addresses.
     >> IPs are allocated on a first-come-first-serve basis.
     >> Config example:
          #access-list 40 permit 192.168.0.240 0.0.0.15
          #ip nat pool NAT_240 196.211.1.116 196.211.1.130 netmask 255.255.255.240
          #ip nat inside source list 40 pool NAT_240

 > OVERLOAD Implementation
     >> Many IL addresses are mapped to ONE IG (public) IP address using different source ports to keep
        track of connections.
     >> Config example:
          #access-list 50 permit 192.168.0.0 0.0.0.255
          #ip nat inside source list 50 interface Dialer0 overload
```

- Outside Source
 > Conceptually just the opposite of Inside Source.
 > The words 'outside source' emphasize the fact that the OG will be changed before entering the network to the OL.
 > Config example:
 >> Traffic from outside host (196.36.75.148) will appear to be coming from a source 10.200.201.1 to local hosts.

```
#ip nat outside source static 196.36.75.148 10.200.201.1 extendable
```

- NAT Timeouts
 > When port translation is configured each entry contains more detail about the traffic that is using it, which gives one finer control over translation entry timeouts.
 > Default values:

>> timeout	- 86,400 sec	(24 hours)
>> icmp-timeout	- 60 sec	(1 minute)
>> udp-timeout	- 300 sec	(5 minutes)
>> tcp-timeout	- 86,400 sec	(24 hours)
>> dns-timeout	- 60 sec	(1 minute)
>> syn-timeout	- 60 sec	(1 minute)
>> finrst-timeout	- 60 sec	(1 minute)

- TCP Load Balancing
 > IP addresses must be contiguous.
 > NAT load-balancing is prone to black-holing traffic, if one of the servers die.

```
CONFIG-SET: NAT Load Balancing
+-----------------------------------
|One old web server replaced by three new servers.
|This allows traffic to be transparently NATed to the new server, without any users' knowing
|
|     access-list 110 permit tcp any host 185.1.1.100 eq www      - 185.1.1.100 is the old web server
|     access-list 110 permit tcp any host 185.1.1.100 eq 443      - It served ports 80, 443, and 8080
|     access-list 110 permit tcp any host 185.1.1.100 eq 8080     - This address is to become a virtual IP
|     !
|     ip nat pool LB 185.1.1.20 185.1.1.22 prefix 25 type rotary  - Defines the new physical servers
|     !
|     ip nat inside destination list 110 pool LB                  - Ties the Virtual IP to the destinations
|     !
|     int fa0/0
|      ip nat inside
|     int s0/1
|      ip nat outside
|
```

- NAT-on-a-Stick
 > Used when a router has only one interface, but translation out of the same interface is still needed.
 > Similar concept to using sub-interface on one physical interface, but with NAT.
 > Done by creating a virtual loopback interface and using PBR (Policy Based Routing).

- NAT Order of Operation
 > The order in which transactions are processed using NAT is based on whether a packet is going from the inside network to the outside network, or from the outside network to the inside network.

```
 > Inside-to-Outside order:
    >>      If IPSec then check input access list.
    >>      Decryption - for CET (Cisco Encryption Technology) or IPSec.
    >>      Check input access list.
    >>      Check input rate limits.
    >>      Input accounting.
    >>      Policy routing.
    >>      Routing.
    >>      Redirect to web cache.
    >>      NAT inside to outside (local to global translation).
    >>      Crypto (check map and mark for encryption).
    >>      Check output access list.
    >>      Inspect (Context-Based Access Control (CBAC)).
    >>      TCP intercept.
    >>      Encryption.
    >>      Queueing.

 > Outside-to-Inside order:
    >>      If IPSec then check input access list.
    >>      Decryption - for CET or IPSec.
    >>      Check input access list.
    >>      Check input rate limits.
    >>      Input accounting.
    >>      NAT outside to inside (global to local translation).
    >>      Policy routing.
    >>      Routing.
    >>      Redirect to web cache.
    >>      Crypto (check map and mark for encryption).
    >>      Check output access list.
    >>      Inspect CBAC.
    >>      TCP intercept.
    >>      Encryption.
    >>      Queueing.

----------
 COMMANDS
----------
# sh ip nat translations [tcp|udp|icmp]          - Shows the active translations
# sh ip nat statistics                           - Shows the NAT statistics
# clear ip nat translation {*|inside|outside}    - Clears the specified translations

#ip nat inside source static {local-ip} {global-ip} [extendable]
                                                 - Creates a static NAT IP-to-IP mapping
#ip nat inside source static [tcp|udp] {local-IP} [local-port] {global-ip} [global-port] [extendable]
                                                 - Creates a static NAT port redirection
#ip nat inside source list {acl} {int|pool} overload
                                                 - Creates overload NAT address translation
#ip nat outside source static {global-ip} {local-ip}
                                                 - Creates a static OG to OL mapping
```

```
#ip nat translation timeout {sec}            - Applies to dynamic translations except for overload translations
#ip nat translation tcp-timeout {sec}        - Applies to the TCP traffic
#ip nat translation udp-timeout {sec}        - Applies to the UDP traffic
#ip nat translation max-entries {entries}    - Limits the maximum number of NAT entries


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=================================*
    NTP (Network Time Protocol)
*=================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Network Management Configuration Guide, Release 12.4T
         > Performing Basic System Management


- There are two ways that a networking device can obtain time information on a network:
 > By polling host servers.
 > By listening to NTP broadcasts.

- Client/Server Protocol
 > The client requests time from server.
 > The clients authenticate servers to validate the source, not the other way around.


- An NTP access-list based restriction scheme allows certain access privileges to be granted or denied.
- NTP uses a 'stratum' or hop count to determine how far away neighboring devices are from the master time source.
- Devices with a lower stratum are considered to be more reliable.
- Switches can't be an NTP server.
- The time can also be configured manually with "clock set" but this will be reset when the system is restarted.
- Manual configuration should only be used as a last resort.
- Optionally, time-zones could be configured.


- Different Polling Modes:
 > Client/Server Mode
      >> The client polls its assigned time serving hosts for the current time.
      >> Client-host relationship, the host will not capture or use any time information sent by the local client device.
      >> Use the "ntp server" command to specify the NTP servers.
 > Peer Mode/Symmetric Active Mode
      >> In this mode the host polls its assigned NTP server for the current time and it responds to polls by its hosts.
      >> A peer-to-peer relationship. The host will also retain time-related information.
      >> Should be used when there is a number of mutually-redundant servers that are interconnected.
      >> Use the "ntp peer" command to specify the NTP peer to consider synchronizing with.
```

```
CONFIG-SET: NTP - Client Authenticating a Server
+-----------------------------------------------------
|Configures the NTP client to authenticate the server's updates.
|
|     ntp authenticate                                      - Enables authentication
|     ntp authentication-key 1 md5 CISCO                    - Defines the authentication keys
|     ntp trusted-key 1                                     - Key numbers for trusted time sources
|     ntp server 142.1.1.6 key 1                            - Configures the client to get time from a server using auth key 1
|
```

```
CONFIG-SET: NTP - Server Authentication Configuration
+-----------------------------------------------------
|     access-list 10 permit 192.168.1.10
|     !
|     ntp master 3                                          - Configures the stratum number. Lower is better!
|     ntp authentication-key 1 md5 CISCO                    - Defines the authentication keys
|     ntp access-group serve-only 10                        - Only allows update to clients matching the ACL
|
```

```
- Broadcast Mode
 > Used when a device wants to receive NTP without asking for it.
 > When a networking device is operating in the Broadcast-Client mode, it does not engage in any polling.
 > Instead, it listens for NTP broadcast packets transmitted by broadcast time servers.
 > Can be a little less accurate though, as the data flows one way.
 > The NTP broadcast server is configured with "ntp broadcast version".
 > The NTP broadcast client is configured with "ntp broadcast client".
```

```
CONFIG-SET: NTP - Broadcast Server and Client Setup
+-----------------------------------------------------
|R1 configured as the broadcast server and R2 as a client
|R1#
|     ntp master 5
|     ntp authentication-key 1 md5 CISCO58
|     !
|     interface FastEthernet 0/0
|     ntp broadcast version 2                               - Enables broadcasting version 2
|     ntp broadcast key 1                                   - Enables broadcasting updates from int fa0/0
|
|R2#
|     ntp authenticate                                      - Enables authentication
|     ntp authentication-key 1 md5 CISCO                    - Defines the authentication keys
|     ntp trusted-key 1                                     - Key numbers for trusted time sources
|     !
|     interface FastEthernet 0/21
|     ntp broadcast client                                  - Enables a client to receive NTP broadcast packets
```

```
- Multicast Mode
 > NTP traffic could be sent more efficiently by using multicast.

CONFIG-SET: Multicasting NTP updates
+--------------------------------------------
|R2#  interface fa0/1
|      ntp multicast 225.0.0.1 ttl 16 version 3            - Configure the NTP multicast server
|      ntp master 2
|
|R1#  interface fa0/0
|      ntp multicast client 225.0.0.1                      - Setup the NTP client to listen for the server
|      ntp multicast version 3
|
>     #show ntp associations [detail]                      - Verify the NTP source, detail and other info
```

```
----------
 COMMANDS
----------
# sh ntp status                                - Shows the status of NTP connections
# sh ntp association [detail]                  - Used to verify NTP associations, authentication

#clock set {hh:mm:ss} {month} {date} {year}    - Manually set the clock, but only valid till a system restart
#clock timezone {zone} {hour} [minute]         - Sets the time zone

#ntp authenticate                              - Enables Authentication, required on all
#ntp authentication-key {number} md5 {value}   - Defines the authentication keys
#ntp trusted-key {key-number}                  - Defines trusted authentication keys, it is required on a client

#ntp server {ip} [ver] [key] [src-int] [prefer] - Used on a client to get time from a server
                                               - [prefer] Specifies a preferred server

#ntp peer {ip} [normal-sync] [ver] [key{id}] [src-int] [prefer]
                                               - For peering devices to get time from each other, based on
                                                   which device has the lower stratum

#ntp source {int}                              - Configures the interface used as source.
#ntp master [stratum]                          - Configuring the System as an authoritative NTP Server
                                               - Lower is better (Default = 7) (value 1-15)
#ntp broadcast [version number]                - Configures the NTP broadcast-server to send NTP broadcasts (Def=3)
#ntp broadcast client                          - Enables a client receive NTP broadcast packets

#ntp access-group {query-only|serve-only|serve|peer} {acl}
                                               - Changes NTP access privileges
                                               - {query-only} Allows only NTP control queries from a peer-system
                                               - {serve-only} Allows only time requests to a peer-system
                                               - {serve} Allows NTP, but only responds to a peer-system
                                               - {peer} Allows the system to synchronize itself with a peer-system
#interface fa0/0
 #ntp disable                                  - Disables NTP services on a specific interface
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===============================================*
     SNMP (Simple Network Managed Protocol)
*===============================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Network Management Config Guides, Release 12.4T
         > SNMP Support

- SNMP uses ports UDP-161 and UDP-162 for traps.
- SNMP is used to report conditions of managed devices to a NMS (Network Management Station).

- The SNMP framework is made up of three parts:
 > SNMP manager          - System that controls and monitors the activities of network hosts.
 > SNMP agent            - The software component within a managed device that reports the requested data, to the managing systems.
 > MIB                   - A database of network management objects, which is used and maintained by the SNMP protocol.

- Message Types
 > GetRequest            - NMS sends this to the agent to retrieve info.
 > GetResponse           - Agent uses this to respond to the NMS.
 > GetNextRequest        - Used by NMS to retrieve the next object instance.
 > SetRequest            - NMS uses this to perform remote config on the agent.
 > Trap                  - Issued by agent to inform the NMS about the change of state of a monitored event.
 > GetBulk               - Allows an agent to respond with chunks of data.
 > Inform                - Allows NMS stations to share trap info.

- SNMP Versions
 > Version 1 uses plain text (default version).
 > Version 2c also uses plain text, but has user authentication and an encrypted password.
 > Version 3 provides the option of encrypting everything.

- IfIndex
 > Each interface gets given an index number at router startup. When the router is reloaded this index number could change.
 > This behavior can be changed with "snmp-server ifindex persist".
 > To see the interface index numbers use "show snmp mib ifmib ifindex".
 > How does a MIB reference this index number?
      >> Example: If a MIB object name of ifEntry.10 is to reference the interface fa2/1 (index 5),
            full MIB object name will be ifEntry.10.5.

- Two ways to collect data:
 > Polling
      >> A NMS system asks managed devices to report on variables.
      >> Uses SNMP community-string, which is a password used by the NMS to poll the device.
      >> Two types of community-strings:
            >>> Read Only - Information gathering only.
            >>> Read Write - Gathers information and can set values.
 > Trapping
      >> Managed devices report events to the NMS.
      >> See configuration steps below.
```

```
- SNMP Community String can be RO/RW/VIEW
 > RO - Allows read access to all MIBs except the community strings themselves.
 > RW - Allows read and write access to all MIBs except the community strings themselves.
```

```
CONFIG-SET: SNMP Polling with a Community-String
+-------------------------------------------------------
|      access-list 2 permit 178.1.2.10                     - Only allow these two hosts to poll the router
|      access-list 2 permit 178.1.2.11
|      access-list 2 deny log                              - Log all other attempts
|      !
|      snmp-server community POLL-READS ro 2               - Specifies a read-only (ro) community, allowing ACL-2's hosts to poll
|      snmp-server community POLL-WRITES rw 2              - Specifies a read-write (rw) community
|

CONFIG-SET: SNMP Traps Example
+-------------------------------------
|      snmp-server enable traps hsrp                       - Enables traps for HSRP only
|      snmp-server location Moon, Planet3.1
|      snmp-server chassis-id 123-98765                    - Configures various SNMP parameters
|      snmp-server system-shutdown                         - Allows router to be reloaded via SNMP
|      !
|      snmp-server trap-source Loopback0                   - Sources traps from Loopback0
|      snmp-server host 185.1.2.200 version 2c MYTRAPS hsrp  - Sends the HSRP Traps to NMS, using version2|
|
```

```
-----------
 COMMANDS
-----------
# sh snmp                                       - Shows the snmp counters
# sh snmp mib ifmib ifindex                     - Shows each interfaces IFINDEX number

#snmp-server community {string} {ro | rw} [acl] - Enables SNMP polling for read-only/read-write
                                                - [acl] Defines who can poll the device

#snmp-server enable traps [notification-type]   - Step-1, Enables all/some snmp traps
                                                - By specifying the type, only specified traps are enabled
#snmp-server host {ip} {community} [type]       - Step-2, Defines an NMS server to trap too
#snmp-server ifindex persits                    - Enables interface ifindex persistence, avoiding the ifindex
                                                      changing after a reboot
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*===============================*
    RMON (Remote Monitoring)
*===============================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Network Management
         > Network Management Config Guide
          > RMON Support

- RMON is used to report an MIB value to a SNMP NMS or syslog server.
- RMON alarms define how an MIB is sampled.

- Two Components:
 > Alarms
      >> The conditions that trigger an event.
 > Events
      >> The messages sent to the NMS/syslog server.

- DELTA Sampling
 > A method to sample the selected variable and calculate the value to be compared against the thresholds.
 > The difference between MIB values at time index A compared to MIB value at time index B.
      >> Number of packets sent out ethernet0/0 each minute.
      >> CRC errors on the interface.
 > Used for any value that is measured as a rate (a value per time).

- ABSOLUTE Sampling
 > Test each sample directly.
 > Exact value of MIB at time index A.
      >> CPU utilization.
      >> Memory utilization.
 > Used for values that increase and decrease.


CONFIG-SET: SNMP RMON Example
+-----------------------------------------------------
|     snmp-server host 123.1.1.1 MYTRAP                     - Sends the MYTRAP traps to the NMS server
|     !
|     rmon event 1 trap MYTRAP desc "CPU above 90%"         - Specifies the rising-threshold event
|     rmon event 2 trap MYTRAP desc "CPU below 30%"         - Specifies the falling-threshold event
|     !
|     rmon alarm 1 lsystem.58.0 60 absolute rising-threshold 90 1 falling-threshold 30 2
|                                                           - Specifies the alarm to watch the CPU processor MIB
|                                                           - Alarm will be triggered if thresholds are exceeded,
|                                                                  and generate the specified events
```

```
-----------
 COMMANDS
-----------
# sh rmon events                                  - Shows the RMON event table
# sh rmon alarms                                  - Shows the RMON alarm table

#snmp-server host {ip} CISCO                      - Enables traps to SNMP server
                                                  - Specifies SNMPv1/v2c community string or SNMPv3 user name

#rmon event {no} (log|desc|trap|owner) (comm)     - [log] Generates a syslog event
                                                  - [trap] Enables trap

#rmon alarm {no} (mib} (sample-rate} (absolute | delta} rising-threshold (value} (event} falling threshold (value} (event}
                                                  - This would use the event above when values match to generate notifications
```

-=-

=====================

Syslog

=====================

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Network Management Config Guide, Release 12.4T
    > Troubleshooting and Fault Management

- Logging level/severity determines which type of log messages will be sent.
- Level 7 (severity) provides the most amount of information, like debugging.
- Level 0 (emergencies) provides the least amount of information.
- Logging at a level will include all the lower levels. If level 3 logging is enabled, levels 2,1 and 0 will be enabled by default.

- Interface-specific events can be logged:

```
 > dlci-status-change    -     DLCI CHANGE messages
 > frame-relay           -     Frame-Relay messages
 > link-status           -     UPDOWN and CHANGE messages
 > subif-link-status     -     Sub-interface UPDOWN and CHANGE messages
```

- Logging History
  > Messages are stored in the history table because SNMP traps are not guaranteed to reach their destination.
  > By default, one message of the level warning and above is stored in the history table even if syslog traps are not enabled.

- Configuration Change Notification (also known as Archiving)
  > DOC-CD LOCATION
   > 12.4T Configuration Guides
    > Configuration Fundamentals Configuration, Release 12.4T
     > Configuration Change Notification and Logging

  > The Configuration Change Notification and Logging feature tracks changes made to the Cisco IOS software running configuration by maintaining a history log.
  > Only complete commands that result in the invocation of action routines are logged.
  > Syntax errors and partially incomplete commands do not get logged.

```
-----------
 COMMANDS
-----------
# sh archive log config{number}                  - Shows first {number} of configuration logs
# sh archive log config statistics               - Shows memory and usage statistics for the config logger
# sh archive config difference nvram:start system:run
                                                 - Compares the difference between the startup & running config
# term [no] monitor                              - Enables/disables the display of log messages to terminal session

#service timestamps {debug|log} {uptime|local}   - Changes the format of the log timestamps
#service sequence-numbers                        - Enable visible sequence numbering of system logging messages
#logging host {ip}                               - Specifies the syslog IP
#logging console {severity}                      - Changes the logging severity (default = 6) for console connections
#logging trap {severity}                         - Limit messages logged to the syslog servers based on severity
#logging facility                                - Controls format of syslog messages
#logging history [level] [size]                  - Changes syslog messages stored in the history file


#archive                                         - Enters archive configuration mode
 #log config                                     - Enters configuration change logger configuration mode
  #logging enable                                - Enables the logging of configuration changes (def = Disabled)
  #logging size {entries}                        - (o) Specifies the maximum number of entries retained (def = 100)
                                                 - When the log is full, oldest entry is deleted with every new entry
  #hidekeys                                      - (o) Suppresses the display of password information in the logs
  #notify syslog                                 - (o) Sends notifications of configuration changes to a remote syslog

#int s0/0
 #logging event {dlci-status-change | frame-relay | link-status | subif-link-status}
                                                 - Enables interfaces specific event logging

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=============================*
   NetFlow
*=============================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS NetFlow Configuration Guide, Release 12.4T
        > Part1: Cisco IOS NetFlow Configuration Guide

- Good article available at http://routing-bits.com/2009/07/14/using-netflow/.
- Netflow captures data from ingress (incoming) and egress (outgoing) packets.
- Instantaneous data can be viewed on the router, or data can be exported to a Netflow interpreter for later analysis.

- A network flow is identified as a unidirectional stream of packets identified as the combination of the key fields below.
- These seven key fields define a unique flow:
 > Source IP address
 > Destination IP address
 > Source port number
 > Destination port number
 > Layer3 protocol type
 > Type of service (ToS)
 > Input logical interface.

- Netflow Top-Talkers
 > The flows that are generating the heaviest system traffic are known as the 'top talkers'.
 > The NetFlow top talkers feature allows flows to be sorted so that they can be viewed by either of the following criteria:
      >> By the total number of packets in each top talker.
      >> By the total number of bytes in each top talker.

-----------
 COMMANDS
-----------
# sh ip flow interface                          - Shows the interfaces which netflow is enabled on
# sh ip cache flow                              - Shows a summary of the netflow statistics, IP's, ports, protocols, etc
# sh ip cache verbose flow                      - Shows a detailed summary of the netflow statistics, including TOS-byte
# sh ip flow top-talkers
# clear ip flow stats                           - Clears the netflow statistics on the router

#interface fa0/3
 #ip flow {ingress | egress}                    - Enables netflow on the interface
                                                - {ingress} Captures traffic that is being received by the interface
                                                - {egress} Captures traffic that is being transmitted by the interface
#ip flow-export dest {ip|hostname} [udp-port]   - Specifies the IP address, or hostname of the netflow collector
#ip flow-export interface-names                 - Export to include the interface names from the flows
#ip flow-export source {int}                    - (o)IP from which interface to be used as a source address
#ip flow-cache entries {number}                 - (o) Changes the number of entries maintained in the netflow cache
#ip flow-cache timeout active {minutes}         - (o) Specifies flow cache timeout parameters for active flows
#ip flow-cache timeout inactive {seconds}       - (o) Specifies flow cache timeout parameters for inactive flows
#ip flow-export ver 9 [origin-as|peer-as][bgp-nh]
                                                - (o)Enables the export of netflow cache entries using the version 9 format
                                                - [origin-as] Export to include the originating AS for the source
                                                      and destination
                                                - [peer-as] Export to include the peer AS for the source and destination
                                                - [bgp-nexthop] Export to include BGP next hop-related information

#ip flow-top-talkers                            - Enters NetFlow Top Talkers configuration mode
 #top {number}                                  - Specifies the maximum number of top talkers
 #sort-by [bytes | packets]                     - Specifies the sort criterion for the top talkers
 #match {class-map|dst|src|protocol|tos}        - Specifies a match criterion


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================================*
  RITE (Router IP Traffic Export)
*=====================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Security Configuration Guide: Securing User Services, Release 12.4T
         > User-Security Configuration
```

```
- The IP Traffic Export feature allows users to configure their router to export IP packets that are received on multiple,
      simultaneous WAN or LAN interfaces.
- The unaltered IP packets are exported on a single LAN or VLAN interface, thereby easing deployment of protocol
      analyzers and monitoring devices.
- IP traffic export eliminates the need for IDS probes to be placed in line, allowing users to place an IDS probe in any
      location within their network or direct all exported traffic to a VLAN that is dedicated for network monitoring.

- Users can configure their router to perform the following tasks:
 > Filter copied packets via an access control list (ACL).
 > Filter copied packets via sampling, which allows the export of one in every few packets of interest.
      Use this option when it is not necessary to export all incoming traffic.
 > Configure bidirectional traffic on an interface (by default, only incoming traffic is exported).

!!NOTE!! Packet exporting is performed before packet switching or filtering.
!!NOTE!! ONLY routed, pass-through traffic is exported; traffic that originates from the router is NOT exported.

----------
 COMMANDS
----------
# sh ip traffic-export [interface | profile]     - Shows information related to exported IP traffic events

# debug ip traffic-export events                 - Enables debugging messages for exported IP traffic packets events

#ip traffic-export profile {profile-name}        - Creates or edits an IP traffic export profile
 #interface fa0/1                                - Specifies the outgoing (monitored) interface for exported traffic
 #bidirectional                                  - (o) Exports incoming and outgoing IP traffic on the monitored interface
 #mac-address {H.H.H}                            - Specifies the MAC of the destination host receiving the exported traffic
 #incoming {ACL | sample one-in-every {number}} - Configures filtering for incoming traffic
 #outgoing {ACL | sample one-in-every {number}} - Configures filtering for outgoing export traffic, (requires bidirectional)

#interface fa0/0                                 - Enter inside interface
 #ip traffic-export apply profile-name           - Enables IP traffic export on an ingress interface


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================*
    IP Accounting
*=====================*
- DOC-CD LOCATION
        > 12.4T Configuration Guides
          > Cisco IOS IP Application Services Configuration Guide, Release 12.4T
           > Configuring IP Services

- Used for the following:
 > To track how many IP packets are received or sent out of an interface,
 > To track how many packets violate an access-list policy configured on an interface, OR
 > Track packets with a certain IP precedence value that are sent or received.

- Optionally, you could limit what IP accounting is kept with a filter.
```

```
----------
 COMMANDS
----------
# sh int s1/0 precedence                        - Verifies precedence accounting
# sh ip accounting access-violations            - Shows access violations in accounting database
# sh ip accounting output-packets               - Shows packets and bytes for a src/dst pair

#ip accounting-list {IP} {wildcard}             - (o) Filters the hosts for which IP accounting information is kept
#ip accounting-threshold {value}                - Specifies the max accounting entries

#interface s0/0
 #ip accounting precedence {input|output}       - Count packets by IP precedence on this interface
 #ip accounting access-violations               - Accounts for IP packets violating access lists on this interface
 #ip accounting mac-address                     - Accounts for MAC addresses seen on this interface
 #ip accounting output-packets                  - Accounts for IP packets output on this interface
```

-=

============================

VTY Access Using Telnet

============================

- DOC-CD LOCATION
      > 12.4TConfiguration Guides
        > System Management
         > Configuration Fundamentals Configuration, Release 12.4T
          > Operating Characteristics for Terminals

- DOC-CD LOCATION (Login-Block)
       > 12.4T Configuration Guides
        > Security and VPN
         > Security Configuration Guide, Release 12.4T
          > Login-Block

- Cisco routers have several VTY lines.
- VTY stands for Virtual TeletType, which is a virtual interface used for telnet, SSH and other types of connections.
- Logging into VTY interfaces are disabled by default, until login enabled and a password is configured.

- IOS Login Enhancements
 > Also known as login block.
 > The login block capability, when enabled, applies to both telnet and SSH connections and more recently to HTTP connections.
 > DOS attack- A malicious user may attempt to interfere by flooding a device with connection requests.
 > Dictionary attack- an attempt to actually gain administrative access to the device.
 > The routing device can be configured to react to repeated failed login attempts by refusing further connection request when login blocking is enabled. This block can be configured for a period of time, called a 'quiet period'.
 > Legitimate connection attempts can still be permitted during a quiet period by configuring an access-list (ACL) with the addresses that you know to be associated with system administrators.

```
CONFIG-SET: IOS Login Enhancements (Login Block)
+-----------------------------------------------------
|     login block-for 100 attempts 15 within 100          - Enables Login Block for 100 sec after 15 attempts within
|     !                                                           a period of 100 sec
|     login quiet-mode access-class ALLOW-R1-R2           - Only R1 & R2 allowed during quiet mode
|     login on-failure log every 10                       - Generates logging messages of failed attempts
|     login on-success log every 15                       - Generates logging messages of successful attempts
|
```

```
- Rotary group
 > By default the VTY numbers are used synchronously. (e.g. first 0, then 1, etc.)
 > Rotary groups allows access to a specific VTY interface. (e.g. always connect the VTY 4)
 > Rotary groups are used by telnetting to an IP on the destination device, using a non-default port (not TCP-23).
 > The Services and Port Numbers used for Rotary Groups differs between platforms. One of these usually work:
     >> 2000, 3000, 6000 and 7000
 > To use one of the service group above, just add the rotary group number. (i.e. 7000+ rotary 12 = port 7012)
```

```
CONFIG-SET: Rotary Group Example
+-----------------------------------------
|R1#  username bob password funny-haha                    - Configures a local username
|     !
|     line vty 0-3                                        - Specifies configuration for the first 4 VTYs
|      login                                              - Enables VTY login
|      password bruce                                     - Configures a VTY password
|     !
|     line vty 4
|     login local                                         - Use the local username/password database
|      rotary 44                                          - Create a rotary group and assigns to VTY 4
|     !
|
|R2#  telnet 10.5.1.1 7044                                - Telnet to a valid IP on R1 using the Rotary group port
|       Trying 10.5.1.1, 7044 ... Open
|       User Access Verification
|
|       Username: bob                                     - Prompted for a username, not just a password 'bruce'
|       Password:
|       R1>
```

```
----------
 COMMANDS
----------
# term [no] monitor                         - Enables/disables the display of log messages to telnet session

#service telnet-zero-idle                   - Sets the TCP window to zero when the telnet connection is idle
#service hide-telnet-address                - Doesn't show the telnet address that's being connected to
#ip telnet hidden                           - Doesn't display telnet addresses or hostnames
#ip telnet quiet                            - Doesn't show anything, like the 'trying...' or 'connecting...'
#ip telnet tos {value}                      - Changes the IPP (default=6) value for locally generated telnet traffic
#login block-for {sec} attempts {no} within {sec}
                                            - Configure IOS login enhancement
```

```
#login quiet-mode access-class {acl}          - (o) Device won't accept any additional connections during quiet period
                                              - Specify what ACL request are allowed during quiet-mode
                                              - If no ACL, ALL requests will be denied during quiet-mode
#login delay {sec}                            - (o) Configures a delay between successive login attempts
#login on-failure log [every {number}]        - (o) Generates logging messages for failed login attempts
#login on-success log [every {number}]        - (o) Generates logging messages for successful login attempts

#line vty 0 4
 #password {pwd}                              - Configures a local VTY password
 #login [local]                               - Specifies the password source, VTY password used by default
                                              - [local] Uses the local username/password database
 #login authentication {group}                - VTY access use AAA login authentication is enabled
 #transport input {none | telnet | ssh | all} - Defines which protocols to use when connecting to the terminal
 #transport output {none | telnet | ssh | all} - Defines which protocols to use for outgoing connections
 #transport preferred none                    - Will prevent the router resolving mistyped commands via DNS
 #busy-message hostname [@ message @]         - Customizes the info displayed during telnet connection attempts
 #vacant-message [@ message @]                - Configures the system to display an idle terminal message
 #refuse-message [@ message @]                - Configures the system to display a "line in use" message
 #private                                     - Saves local settings between sessions
 #length {screen-length}                      - Sets the screen length
 #width {characters}                          - Sets the screen width
 #session-limit {number}                      - Sets the maximum number of simultaneous sessions
 #lockable                                    - Enabling session locking
 #ip tcp chunk-size {number}                  - Optimizes the line by setting the number of characters-output,
                                                    before the interrupt

 #ip alias ip-address {tcp-port}              - Assigns an IP address to the service provided on a TCP port.
 #service {linenumber}                        - Displaying line connection information after the login prompt

 #escape-character {ascii|break|default|none} - Changes the system escape character (def= Ctrl-Shift-6, X)
                                              - {default} To restore the default escape sequence

                                              >>> VTY Timeouts <<<
 #exec-timeout {minutes} {seconds}            - Used to end an idle exec process. To disable set the value = 0
 #absolute-timeout {minutes} {seconds}        - Will end an exec process, after timer expires, even if not idle
 #logout-warning {seconds}                    - A warning is displayed prior to the user being logged out


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================================*
  VTY Access using SSH (Secure Shell)
*======================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Security Configuration Guide: Securing User Services, Release 12.4T
        > Secure Shell (SSH)
```

- The SSH server feature enables a SSH client to make a secure, encrypted connection to a Cisco router.
- This connection provides functionality that is similar to that of an inbound telnet connection, but is secure.

- The SSH integrated client feature is an application running over the SSH protocol to provide device authentication and encryption.
- The SSH client enables a Cisco router to make a secure, encrypted connection to another Cisco router or to any other device running the SSH server.

```
----------
 COMMANDS
----------
# sh ip ssh                                 - Shows if enabled, the version and configuration data
# sh ssh                                    - Shows the status of the SSH server connections
# ssh -l {username} {ip}                    - Used to test a SSH connection to a SSH host,
                                            - {-l} Optional- Used to specifies a username

#username name [privilege level] {password} - Establishes a local username-based authentication database
#hostname {HOSTNAME}                        - Specifies a router hostname
#ip domain-name {NAME}                      - Creates a domain name
#crypto key generate rsa                    - Generates the RSA pair-keys using the hostname.domain-name

#ip ssh version 2                           - (o) Enables version 2
#ip ssh time-out {seconds}                  - (o) This setting applies to the SSH negotiation phase (def = 120sec)
#ip ssh authentication-retries {number}     - (o) Specifies the number of authentication retries (def = 3)

#line vty 0 4
 #transport input ssh                       - Changes the transport input to SSH and set the login type
 #login local                               - At login to use local username database
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
    SCP (Secure Copy)
*========================*
```

- DOC-CD LOCATION
    > 12.4T Configuration Guides
     > Cisco IOS Security Configuration Guide: Securing User Services, Release 12.4T
      > SSH (Secure Shell)
       > SCP (Secure Copy)

- The SCP feature provides a secure and authenticated method for copying router configuration or router image files.
- SCP relies on SSH, an application and a protocol that provides a secure replacement for RCP.

- Before enabling SCP, SSH must be configured correctly on the router (refer to Section above).
- SCP also requires that AAA authorization be configured so the router can determine whether the user has the correct privilege level.

```
----------
 COMMANDS
----------
# debug ip scp                              - Troubleshoots SCP authentication problems

#aaa new-model                              - Enables the AAA access control system
```

```
 #aaa authentication login default local        - Sets AAA authentication at login to use local username database
 #aaa authorization exec default local          - Sets parameters that restrict user access to a network
                                                - {exec} Determines if the user is allowed to run an EXEC shell

#username name [privilege level] (password}     - Establishes a local username-based authentication database
#ip scp server enable                           - Enables SCP server-side functionality

#scp {file-location/filename} {username@IP:filename}  - SCP an IOS file to a router



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=================*
   Banners
*=================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4T
         > Part 4: Banners and Menus

- Banners types:
 > motd                 - First banner displayed before login prompt.
 > login                - After MOTD but before login prompt.
 > exec                 - After login prompt once "exec" is invoked.
 > incoming             - Reverse telnet banner when opening connection.
 > busy                 - Specifies a local message globally when telnetting to a server and it is busy.
 > prompt-timeout       - Sets message for login authentication timeout.


- Configuring banners using tokens to display special argument:
 > $(hostname)          - Router host name.
 > $(peer-ip)           - IP address of the peer machine.
 > $(gate-ip)           - IP address of the gateway machine.
 > $(domain)            - Shows the domain name for the router.
 > $(line)              - Shows the VTY or TTY (asynchronous) line number.
 > $(line-desc)         - Shows the description attached to the line.

----------
 COMMANDS
----------
#banner [exec|incoming|login|motd|prompt-timeout]
                                                - Configures the specified banner

#line {con|vty|aux} {number}
 #no motd-banner                                - Disables the MOTD banner display only on the specified line method
 #no exec-banner                                - Disables the MOTD and EXEC banner display on the specified line method
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*====================*
   IOS Menus
*====================*
- Used to present a configurable menu with limited commands.
- The menu could be manually invoked or the "autocommand" could be used for a VTY or a user.
- Refer to this link for detailed use: http://routing-bits.com/2008/09/30/cisco-terminal-server-with-menu-command/.
- There are four portions to the menu:
 > menu title           - Creates a title (banner) for a user menu.
 > menu text            - Specifies the text of the items in a user menu.
 > menu prompt          - Specifies the prompt for a user menu.
 > menu commands        - Specifies underlying commands for user menus.

CONFIG-SET: Configures a Custom IOS Menu
+-----------------------------------------------------------
|     menu TS title @ MY RACK @                         - Specifies the menu title
|     menu TS text 1 Go to R1                           - Specifies the text for the menu items '1' = command 1
|     menu TS text 2 Go to R2
|     menu TS text s show all sessions                  - 's' specifies the command s = "show sessions"
|     menu TS text c<no> clear the sessions             - 'c<no>' specifies the command alias 'c11'
|     menu TS text e menu-exit
|     menu TS text q Quit terminal server session
|     menu TS prompt [d title d]                        - Specifies the menu prompt text
|     menu TS command 1 resume R1 /connect telnet R1    - Specifies the command to be performed when the menu item is selected
|     menu TS command 2 resume R2 /connect telnet R2    - I.e. if 2 is pressed, telnet to R2 or resume the connection if exists
|     menu TS command 3 resume R2 /connect telnet R3
|     menu TS command e menu-exit                       - Allows an option to exit from the menu
|     !
|     menu TS command s show sessions                   - Executes the command "show sessions"
|     menu TS options s pause                           - Pause required to display the output and wait for user input
|     !
|     menu TS command c1 disconnect 1                   - Disconnects session 1
|     menu TS command q exit                            - Quits from terminal-server completely
|     !
|     line vty 0 4
|      autocommand menu TS                              - Invokes the menu for every access to VTY line 0 to 4

----------
 COMMANDS
----------
# menu {name}                                   - Invokes a user menu from Exec Mode

#menu title                                     - Creates a title, or banner, for a user menu
#menu prompt {text}                             - Specifies the prompt for a user menu
#menu text {text}                               - Specifies the text of a menu item in a user menu
#menu {name} command {item} {command}           - Specifies the underlying commands of a menu
#menu {name} command {item} {menu-exit}         - Specifies a way to exit a menu
#menu {name} options {options}                  - Sets options for items in user menus
#menu {name} clear-screen                       - Clears the terminal screen before displaying a menu
#menu line-mode                                 - Requires the user to press Enter after specifying an item
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   HTTP Server
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Network Management Configuration Guide, Release 12.4T
         > HTTP Services

----------
 COMMANDS
----------
#ip http server                                 - Enables the HTTP 1.1 server, including the Cisco web browser interface
#ip http secure-server                          - Enable secure HTTP, requires standard http server to be disabled
#ip http auth {aaa|enable|local|tacacs}         - (o) Specifies the authentication method to be used for login
#ip http path url                               - (o) Sets the base HTTP path for HTML files
#ip http access-class access-list-number        - (o) Limits access to the HTTP server
#ip http max-connections value                  - (o) Sets the max concurrent connections


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    TFTP Server
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4T
         > Part7: Configuring Basic File Transfer Services

- Order of image file booting:
 > When a router boots up, it will look in its global config for each "boot" command and then try them sequentially.
 > If there aren't any boot commands specified, the router will fail over to using the first valid image in flash.
 > If no valid image is found, the router will then try to boot a default image using TFTP. The default IOS image name is hardware
     dependent and can be seen during ROMMOM mode by issuing the command "confreg".
 > Example the default boot image name for 2600 is 'cisco2-c2600;.

----------
 COMMANDS
----------
#boot system flash {file-name}                  - Specifies the first boot option to be used
#boot system tftp {file-name} {tftp-server-ip}  - Specifies the second boot option to be used
                                                - Specifies that a client router load a system image from a TFTP-server
#boot system rom                                - Specifies the third boot option to be used
                                                - Specifies that the client router loads its own ROM image
                                                       if the load from a server fails
#tftp-server flash:{file-name} [alias] [acl]    - On a cisco router as TFTP-server specify image location
                                                - [alias] Used to alias default IOS image names
                                                - [ACL] Access list of requesting hosts allowed
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*======================*
    FTP Server
*======================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4T
         > Part7: Configuring Basic File Transfer Services

- FTP can also be used to transfer files between systems on the network.
- FTP is more preferable than TFTP, because of a higher file-transfer rate.
- For large IOS upgrades use FTP, e.g. try and copy a 100MB image across the network and see how long TFTP takes.

----------
 COMMANDS
----------
#ip ftp username {name}                          - Sets the required FTP username
#ip ftp password {pwd}                           - Sets the required FTP password
#ip ftp passive                                  - Configures the router to only use passive-mode FTP connections
#ip ftp source-interface {int}                   - Specifies the source IP address for FTP connections


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================================*
   CDP (Cisco Discovery Protocol)
*=====================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Network Management Configuration Guide, Release 12.4T
         > Cisco Discovery Protocol (CDP)

- For a Frame-Relay encapsulated interface, CDP is not enabled by default on the physical interface, only on the sub-interface.

----------
 COMMANDS
----------
# sh cdp                                         - Shows all the CDP protocol info
# sh cdp interface                               - Shows CDP enabled interfaces
# sh cdp entry {device}                          - Shows information about a specific neighbor
# sh cdp traffic                                 - Shows CDP counters and traffic

# clear cdp table                                - Deletes the CDP table of information about neighbors

#cdp timer 30                                    - Changes the CDP timer (default=60)
#cdp holdtime 90                                 - Changes the CDP hold timer interval (default=180)
#no cdp run                                      - Disables CDP on a supported device (default=enabled)
#int fa0/0
 #no cdp enable                                  - Disables CDP on a supported interface (default=enabled)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*========================================*
   WCCP (Web Caching Content Protocol)
*========================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Application Services Configuration Guide, Release 12.4T
         > Configuring WCCP

- The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that intercepts IP packets and
      redirects those packets to a destination other than that specified in the IP packet.
- Typically the packets are redirected from their destination web server on the Internet to a content engine that is
      local to the client.
- WCCP enables you to integrate content engines into your network infrastructure.
- WCCP works only with IPv4 networks.

- There are two versions.
 > WCCPv1 supports the redirection of HTTP traffic only.
 > WCCPv1 does not allow multiple routers to be attached to a cluster of content engines.
 > WCCPv2 supports service groups that can comprise up to 32 content engines and 32 routers.
 > WCCPv2 is the default version.

- 3550 Switches.
 > To support WCCP the SDM profile needs to be changed to EXTENDED-MATCH.
 > Supports only inbound redirection.

- WCCP uses UDP-2048 and GRE.

 ----------
 COMMANDS
 ----------
# sh sdm prefer                                - Shows the current SDM template and stats
# sh ip wccp interface

#sdm prefer extended-match                     - Required on Cisco-3350 to enable support of WCCP

#ip wccp version {1 | 2}                       - (o) Changes the version (default = 2)

#ip wccp web-cache [group-list] [redirect-list] - Enables WCCP
                                               - [group-list] Limits the content engines permitted to participate
                                               - [redirect] Limits what requests are redirected
#interface fa0/0
 #ip wccp web-cache redirect {in|out}          - Enables WCCP on an interface
                                               - {in|out} Specifies direction to listen for http requests
 #ip wccp redirect exclude in                  - (o) Excludes traffic on the specified interface from redirection
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=============================*
    IP and Command Aliases
*==============================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Additional and Legacy Protocols
         > Cisco IOS Terminal Services Configuration Guide, Release 12.4T
          > Configuring Dial-In Terminal Services

- IP Alias
 > Used to assign an IP address to a service provided on a TCP port.
 > The IP address must be on the same network or subnet as the main address of the terminal server,
      and must not be used by another host on that network or subnet.
 > Connecting to the IP address has the same effect as connecting to the main address of the router,
      using the argument tcp-port as the TCP port.

- Aliases
 > command alias allows alternative or shorter syntax for a command to be configured.
 > Examples:
      #alias exec SEN send *
      # alias exec CL clear interface counters
      # alias exec sib show ip interface brief
 > Don't confuse command 'alias' with 'ip alias'.

 -----------
 COMMANDS
 -----------
#ip alias {IP} {tcp-port}                         - Specifies the IP for the service on the TCP Port
#alias mode {name} {command-line}                 - Configures a command alias



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===========================*
    IP Event Dampening
*===========================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4T
         > Configuring IP Routing Protocol-Independent Features

- Configures a router to automatically dampen a flapping interface (use the dampening command in the interface configuration mode).
- Can also be used to suppress IGP advertisement of the interfaces after router reload with the "restart-penalty".

- The IP event dampening feature will function on a sub-interface but cannot be configured on only the sub-interface.
- Only the primary interface can be configured with this feature. Primary interface configurations are
      applied to all sub-interfaces by default.
```

```
- Optional Timers
 > Half-life-period
      >> Once the route has been assigned a penalty, the penalty is decreased by half after the half-life period expires.
      >> The default time is 5 sec.
 > Reuse-threshold
      >> When the accumulated penalty decreases enough to fall below this value, the route is unsuppressed.
      >> The default is 1000.
 > Suppress-threshold
      >> A route is suppressed when its penalty exceeds this limit.
      >> The default is 2000.
 > Max-suppress-time
      >> Maximum time (in seconds) a route can be suppressed.
      >> The default is four times the half-life-period value (20 sec).
 > Restart-penalty
      >> Penalty to applied to the interface when it comes up for the first time after the router reloads.
      >> The default is 2000 penalties.

----------
 COMMANDS
----------
# clear counters                                - Clears the interface counters
# sh dampening interface                        - Shows a summary of interface dampening
# sh interface dampening                        - Shows a summary of the dampening parameters and status

#dampening [half-life} [reuse] [suppress] [max-suppress-time] [restart-penalty]
                                                - Configures a router to automatically dampen a flapping interface
                                                - See the values above


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Crash Dump
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Cisco IOS Network Management Configuration Guide, Release 12.4T
         > Troubleshooting and Fault Management

- A crash dump is used for analysis when a router has crashed, to find the root cause.
- If using FTP, a username and password must be configured for FTP.
- If the destination dump IP is not on a directly-connected link, the "ip default-gateway" command is required.

----------
 COMMANDS
----------
#ip ftp username {name}                         - Sets the required FTP username
#ip ftp password {pwd}                          - Sets the required FTP password
#exception core-file {name}                     - Sets name of core dump file
#exception protocol {ftp|rcp|tftp}              - Sets protocol for sending core file. FTP should be preferred
#exception dump {ip}                            - Sets name of host to dump to
```

Warm Reload

- DOC-CD LOCATION
  > 12.4 T Configuration Guides
   > Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4T
    > Part 9: Loading and Maintaining System Images
     > Warm Reload

- In typical confusion Cisco calls this feature warm reload, but the command used is "warm-reboot".
- The router saves initial data (as stored in IOS image) in a separate memory region.
- Then it reuses the saved data together with the IOS code already residing in RAM to restart the IOS.
- Of course, the IOS code (depending on the platform's memory management capabilities) or saved data could get corrupted.
- Therefore the warm reload cannot be used continuously.
- The router will fail back to a traditional reload if the router crashes before the specified time interval.
- One cool thing about this is that a router can be warm-rebooted without its flash card (cold-reboot will not work).

!!NOTE!! After a warm reboot is enabled, it will not become active until after the next cold reboot because a warm reboot requires a copy of the initialized memory!!

COMMANDS

```
# sh warm-reboot                               - Shows statistics for attempted warm reboots
# sh region | i saved                          - Shows the amount of memory used and address blocks

# reload warm {in | at | cancel}               - Allows a reload without losing the warm-boot configuration

#warm-reboot [count number] [uptime minutes]    - Enables a warm-reboot
                                               - [count] Number of warm reboots allowed between cold-reboots. (def=5)
                                               - [uptime] Minimum time to lapse after initial boot before warm-reboot
                                                     will be enabled
```

System Resources

- DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Cisco IOS Network Management Configuration Guide, Release 12.4T
    > System Monitoring and Logging

- CPU Threshold Notifications
  > Notifies users when a predefined threshold of CPU usage is crossed by generating an SNMP trap message for the top users of the CPU.
  > Two types of CPU utilization threshold are supported:
    >> Rising Threshold

```
            - A rising CPU utilization threshold specifies the percentage of CPU resources used, when exceeded for a
                  configured period of time, before a notification is issued.
        >> Falling Threshold
            - A falling CPU utilization threshold specifies the percentage of CPU resources used, when CPU usage falls
                  below this level for a configured period of time, before a notification is issued.
 > Requires SMTP to be configured (refer to the SNMP Section).

- Memory Threshold Notifications and Reservation
 > There are two ways to mitigate low-memory conditions on a router:
      >> Threshold notifications can be sent to indicate that free memory has fallen below a configured threshold.
      >> Memory reservation can be configured to ensure that sufficient memory is available to issue critical notifications.
```

```
 > Here are two examples of the threshold notifications generated:
   1- If the available free memory is less than the specified threshold:
      000029: *Aug 12 22:31:19.559: %SYS-4-FREEMEMLOW: Free Memory has dropped below 2000k
      Pool: Processor  Free: 66814056  freemem_lwm: 204800000

   2-If the available free memory has recovered to more than the specified threshold:
      000032: *Aug 12 22:33:29.411: %SYS-5-FREEMEMRECOVER: Free Memory has recovered 2000k
      Pool: Processor  Free: 66813960  freemem_lwm: 0

 > Memory reservations: The amount of memory reserved for critical notifications may not exceed 25% of total available memory.

CONFIG-SET: CPU and Memory Thresholding Example Question
+-----------------------------------------------------------
|The router should generate a log message when total CPU usage is above 50% with the smallest possible sampling interval
|Additionally, it should log a syslog message when its free processor memory falls below 1 Mbyte and reserve 512 Kbytes of
|memory for the notification process itself.
|
|     process cpu threshold type total rising 50 interval 5       - If the CPU threshold rises above 50%, for more than
|     !                                                                 5 sec, then trigger a notification
|     memory free low-watermark processor 1000                    - Specifies a threshold of 1000KB of free processor
|     memory reserve critical 512                                 - Reserves 512KB of memory

----------
 COMMANDS
----------                                          >>> CPU Threshold Notifications<<<
#snmp-server enable traps cpu threshold             - Enables CPU thresholding notification as traps and inform requests
#snmp-server host {IP} traps {public cpu}           - Sends CPU traps to the specified address

#process cpu threshold type {total} rising {%} interval {seconds} falling {%} interval {seconds}
                                                    - Sets the CPU thresholding notifications types and values

                                                    >>> Memory Threshold Notifications <<<
#memory free low-watermark {processor | io}         - Specifies a threshold in kilobytes of free processor or I/O memory
#memory reserve critical {kilobytes}                - Reserves memory in kilobytes for the issue of critical notifications
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==================================*
   KRON Command Scheduler
*==================================*
- Allows exec commands or TCL scripts to run at a specific time.
- KRON configuration consists of a policy-list containing exec commands and a scheduler to execute the commands in the
      policy-list at a specific time or recurring interval.

----------
 COMMANDS
----------
# sh kron schedule                            - Shows the status and schedule of occurrences.

#policy-list {policy-name} [conditional]      - Defines a policy-list
                                              - [conditional] Execution will stop on failure return values
 #cli {command string}                        - List the commands to be run

#kron occurrence {NAME}{in|at}{hh:min}[one|recurring} - Creates a KRON occurrence
 #policy-list {policy-name}                   - Attach the policy-list to execute


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==================================*
   EEM (Embedded Event Manager)
*==================================*
- DOC-CD LOCATION
       > 12.4 T Configuration Guides
        > Cisco IOS Network Management Configuration Guide, Release 12.4T
         > EEM (Embedded Event Manager)

- EEM is a great way for those who love scripting and automation to make their networking devices do some interesting things.
- EEM was designed to offer event management capability directly on Cisco IOS devices.
- EEM offers the ability to monitor events and take informational or corrective action when the monitored events occur or
      when a threshold is reached.
- An EEM policy is an entity that defines an event and the actions to be taken when that event occurs.
- There are two types of EEM policies: an applet and a script.
 > Applet
      >> An applet is a simple form of policy that is defined within the CLI configuration.
 > Script
      >> A script is a form of policy that is written in TCL (Tool Command Language).

- EEM uses software programs known as event detectors to determine when an EEM event occurs.
- Some of the notable Event Detectors (availability depends on IOS) are:
 > CLI Event Detector                   - Detects various CLI command types based on regular expressions.
 > IP SLA Event Detector                - Detects when an IP SLA reaction is triggered.
 > NetFlow Event Detector               - Detects when a NetFlow event is triggered.
 > None Event Detector                  - Used when "event manager run" CLI command executes an EEM policy.
 > Interface Counter Event Detector     - Responds to interface counters crossing thresholds.
 > Routing Event Detector               - Detects events when a route entry changes in the RIB (Routing Information Base).
 > SNMP Event Detector                  - Allows a standard SNMP MIB object to be monitored.
 > SNMP Notification Event Detector      - Intercepts SNMP trap and inform messages coming into or going out of the router.
 > Syslog Event Detector                 - Screening of syslog messages with a regular expression pattern match.
 > Watchdog                              - Generates periodic timer events and allows the EEM applets to be repeated.
```

- Embedded Event Manager Actions
 > The CLI-based corrective actions that are taken when event detectors report events. Enable a powerful on-device event management mechanism.
 > Event action availability depends on the IOS release.
 > Example of some of the actions (refer to the EEM built-in environment variables):
     >> Executing a CLI command.
     >> Sending a short e-mail.
     >> Reloading the Cisco IOS software.
     >> Generating an SNMP trap.
     >> Setting the state of a tracked object.
 > EEM action CLI commands contain an EEM action label that is a unique identifier.
 > Actions are sorted and run in ascending alphanumeric key sequence using the label.
 > If using numbers as labels, be aware that alphanumerical sorting will sort 10.0 after 1.0, but before 2.0.
 > So rather make use of the numbers, as I did in the config-set below.

- See this post on how to get your Cisco router to log events/outputs to Twitter.
     http://routing-bits.com/2010/01/20/tweeting-router/.

- EEM Environment Variables
 > By convention, all Cisco EEM environment variables begin with "_" (underscore).
 > E-mail-specific environmental variables:

```
     >> _email_server  - The e-mail server name (username:password@host format is allowed).
     >> _email_to            - The address to which e-mail is sent.
     >> _email_from          - The address from which e-mail is sent.
     >> _email_cc            - The address to which the e-mail is copied.
```

CONFIG-SET: EEM applet- Preventing a Loopback Interface From Being Shutdown

```
+-------------------------------------------------------------------------------
|     event manager applet Lo0                             - Creates and registers the applet with EEM
|      event syslog occurs 2 pattern "Loopback0.*down"     - Configures a syslog event detector to match the interface message
|      action 1.0 syslog msg "The loopback0 went down"     - Configures a syslog message detecting the event
|     !
|      action 1.1 cli command "enable"                     - Configures actions to be taken
|      action 1.2 cli command "configure terminal"
|      action 1.3 cli command "int lo0"
|      action 1.4 cli command "ip add 10.1.1.1 255.0.0.0"
|      action 1.5 cli command "no shut"
|      action 1.6 syslog msg "THIS WILL BE REPORTED"
|     !
|      action 1.7 cli command "show users"                 - Sees who is logged on to the router
|     !
|     !                                              - The next command initiates an email including the previous command output
|      action 1.8 mail server "10.1.1.1" to "me@bob.com" from "test@lab.com" subject "lo0"
|                 body "someone is playing as per $_cli_result"
|
----------
 COMMANDS
----------
# sh event manager environment                 - Shows the EEM environment variables set
# sh event manager detector <name> [detail]    - Shows the variables detector
# sh event manager policy registered           - Shows the EEM policies that are currently registered
# sh event manager history events              - Shows detailed information about each EEM events
# sh event manager history traps               - Shows the EEM SNMP traps that have been sent

# debug event man action cli                   - Enables EEM CLI event debugging
# debug event man action mail                  - Enables EEM action email debugging
# tclsh flash:tcl/clear10.tcl                  - References a TCL script in flash:

# event man run <applet>                       - Manually executes a none-event applet (requires event set to none)

#event manager environment {variable string}    - Configures the value of the specified EEM environment variable
#event manager directory user policy {path}    - Defines the location where the user-defined TCL script is stored
#event manager policy {name.tcl} [type {system|user}]
                                               - Registers the EEM TCL script

#event manager applet <name>                   - Creates and registers the applet with EEM
 #event {detector} {string options}            - Specifies the event criteria that cause the EEM applet to run
 #action {label} {type} {string options}       - Specifies the action when an EEM applet is triggered

#event manager scheduler suspend               - Immediately suspends the execution of all EEM policies



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=========================*
   Other Services
*=========================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
         > Cisco IOS IP Application Services Configuration Guide, Release 12.4T
          > Configuring TCP

- The features which comply with RFC 1323, TCP Extensions for High Performance, are:
 1- TCP Selective Acknowledgment
      > This feature improves performance in the event that multiple packets are lost from one TCP window of data.
 2- TCP Time-Stamp
      > Provides better TCP round-trip time measurements.
 3- TCP Window Scaling
      > A larger window size is recommended to improve TCP performance in network paths with large bandwidth-delay product
            characteristics, called Long Fat Networks (LFNs). The TCP Window Scaling enhancement provides that support.
      > The window scaling extension in the Cisco IOS software expands the definition of the TCP window to 32 bits and then
            uses a scale factor to carry this 32-bit value in the 16-bit window field of the TCP header.
      > The default TCP window size is 4128 bytes if window scaling was not configured.
 4- TCP ECN
      > Provides a method for an intermediate router to notify the end hosts of impending network congestion.
```

- TCP Keepalive
 > The TCP keepalive capability allows a router to detect when the host with which it is communicating experiences a system failure, even if data stops being sent (in either direction).
 > Keepalives are sent once every minute on otherwise idle connections. If five minutes pass and no keepalives are detected, the connection is closed.
 > If the host replies to a keepalive packet with a reset packet, the connection is also closed.

- TCP Synwait-Time
 > Is the amount of time the Cisco IOS software will wait for a TCP connection to be established.
 > It does not pertain to traffic going through the device, just to traffic originated by the device.
 > Configured with "ip tcp synwait-time".

- TCP Small Services
 > By default, the TCP services for Echo, Discard, Chargen and Daytime are disabled.
 > These services are used to test the TCP transport functionality.

- UDP Small Services
 > By default the UDP services for Echo, Discard and Chargen are disabled.
 > These services are used to test the UDP transport functionality.

- Service Nagle
 > A standard TCP implementation sending keystrokes between machines will tend to send one packet for each keystroke typed, which can be less optimal.
 > John Nagle's algorithm (RFC 896) helps alleviate the small-packet problem in TCP.
 > The first character typed after connection establishment is sent in a single packet, but TCP holds any additional characters typed until the receiver acknowledges the previous packet.
 > Then the second, larger packet is sent, and additional typed characters are saved until the acknowledgment comes back.
 > The effect is to accumulate characters into larger chunks, and pace their transmission to the network at a rate matching the round-trip time of the given connection. This method is usually preferable for all TCP-based traffic.

- Scheduler Allocate
 > Allows some measure of control in apportioning a router CPU between interrupt processing vs. process mode.
 > With "no scheduler allocate", the interrupt processing can use 100% of the CPU and entirely lock-out any process context activity.
 > "scheduler allocate 3000 1000" is typically a decent setting.

- DRP Server Agent
 > A DRP server agent is a border router or peer to a border router that supports the geographically distributed servers for which Distributed Director service is desired.
 > Distributed Director makes decisions based on BGP and IGP information, meaning that all DRP server agents must have full access to BGP and IGP routing tables.

 COMMANDS

```
# show tcp brief [all]                    - Shows an established and listen TCP connection currently on the router
# clear tcp                               - Clearing non-functioning TCP connections
# clear tcp {local|remote}                - Terminates the specific TCP connection identified

#ip tcp selective-ack                     - Enables TCP selective acknowledgment
```

```
#ip tcp timestamp                            - Provides better TCP round-trip time measurements
#ip tcp window-size {size-bytes}             - Configure the TCP window size. (default = 4128 bytes)
#ip tcp ecn                                  - Enables ECN for TCP
#service {tcp-keepalives-in|tcp-keepalives-out} - Generates TCP keepalive packets on idle network connections
#ip tcp synwait-time {sec}                   - Changes the time a router will wait for establishing TCP connections
                                                   (including telnet/SSH) coming from the router (default = 30 sec)

#service tcp-small-servers                   - Enables the TCP small servers
#service udp-small-servers                   - Enables the UDP small servers
#scheduler allocate {interrupts} {processes} - Allows some control between interrupt processing vs. process mode
#service nagle                               - Enables the Nagle slow packet avoidance algorithm
#ip drp server                               - Enables a DRP server agent
#ip drp access-group {acl}                   - Controls the sources of valid DRP queries by applying a standard ACL
```

-=-

=====================================

Disabling Unnecessary Services

=====================================

- Source Routing
 > Allows the source to determine the route the packet will take through the network to reach the destination.
 > Two types of source routing:
     >> Loose   - The complete route is not included in the packet, and can take any path through the network to reach the destination.
     >> Strict - The packet must only pass through the defined routers listed in the header of the packet to reach the destination.
 > Possible security risk, but can also be used for troubleshooting, using the telnet, ping, or trace on the Cisco IOS.
 > Enabled by default.

- BOOTP and DHCP
 > BOOTP was developed long before DHCP and is enabled by default.
 > BOOTP is disabled with "no ip bootp".
 > Even when BOOTP is disabled, the router will still listen on UDP-67 if DHCP is enabled.
 > DHCP is disabled with "no service dhcp".

- CDP
 > Although CDP is a great aid in troubleshooting (refer to CDP Section) it can be a potential security risk if enabled on the wrong interfaces.
 > Enabled by default.

- Finger
 > Used to see what users are logged on to the network device.
 > The "service finger" command has been replaced by the "ip finger" command to disable the service.
 > Enabled by default.

- IP ICMP Redirect
 > An ICMP redirect message can be generated by a router when a packet is received and transmitted on the same interface.
 > The router will then forward the original packet and send a ICMP redirect message back to the original sender.
 > This behavior allows the sender to bypass the router and forward future packets directly to the destination.
 > Enabled by default.

- Proxy ARP
  - Proxy ARP enables a router to respond with its own interface MAC if a host is trying to reach another host on a different subnet, and the router has a valid entry in the routing table for that destination host.
  - Disabling Proxy-ARP can cause complications, especially with default routing.
  - When proxy ARP is disabled, for each destination the router will try to find the layer3-to-layer2 mapping.
  - Enabled by default.

- IP-Unreachables
  - Used to enable the generation of ICMP-unreachable messages.
  - When a traceroute probes time out (TTL=0), by default a router responds with an IP-unreachable message.
  - The command "no ip unreachables" under an interface disables that ICMP response.
  - This disable command is often used to hide network devices in trace-routes.
  - Enabled by default.

- IP Mask-Reply
  - Responds to ICMP mask requests by sending out ICMP mask replies, and these mask replies contain important network information.
  - Disabled by default.

- IP Directed Broadcast
  - A service that is commonly used in Smurf attacks.
  - Smurf attacks send ICMP echo requests from a spoofed source address to a directed broadcast that cause all hosts to respond to the ping echo request.
  - This creates a lot of traffic on the network, often undesirable.
  - Disabled by default.

COMMANDS

```
# sh ip redirects                               - Shows the hosts for which an ICMP redirect message has been received

#no ip source-route                             - Disables source-routing options
#no ip bootp                                    - Disables (BOOTP) bootstrap server
#no service dhcp                                - Disables the DHCP service
#no cdp run                                     - Disables CDP globally
#no ip finger                                   - Disables the finger command
#no service finger                              - Disables the finger command (older syntax)
#no ip icmp redirect                            - Disables IP ICMP redirect globally
#interface fa0/0
 #no ip redirects                               - Disables IP ICMP redirect messages on an interface
 #no ip proxy-arp                               - Disables proxy ARP
 #no cdp enable                                 - Disables CDP for the interface
 #no ip unreachables                            - Prevent the interface from generating unreachables
 #no ip mask-reply                              - Disables IP mask-reply if previously enabled
 #no ip directed-broadcast                      - Disables IP Directed Broadcast if previously enabled
```