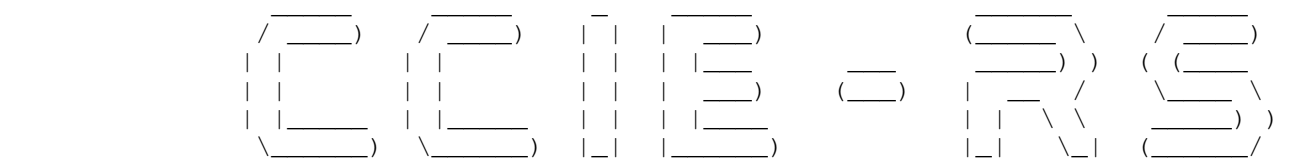
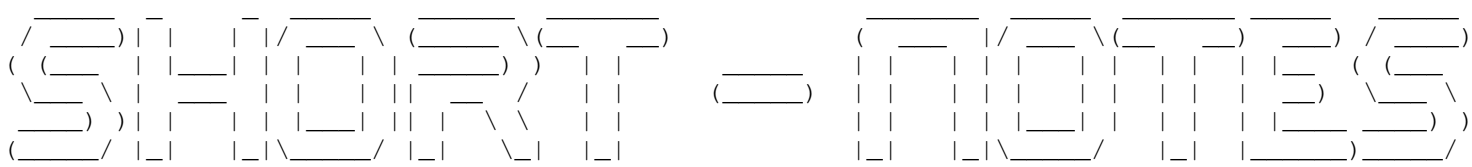




Version 4.2
(Includes Troubleshooting)

Written and Compiled by Ruhann du Plessis
CCIE R&S 24163

Routing-Bits.com

-=-

COPYRIGHT INFORMATION

-=-

CCIE Short-Notes v4
by Ruhann Du Plessis
CCIE R&S #24163, CCNP, CCIP.
http://www.routing-bits.com
http://blog.ru.co.za

Version 4.2

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

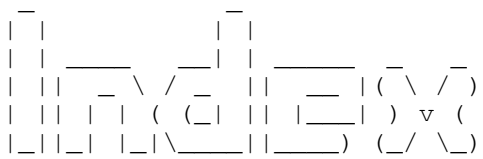
DISCLAIMER

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

This publication, CCIE Short-Notes v4 is designed to provide technical information and assist candidates in the preparation for CISCO Systems CCIE Routing and Switching Lab Exam. The information can also assist any networking engineer in day-to-day duties.

While every effort has been made to ensure this book as 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 belongs 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.



MOTIVATION FOR THIS BOOK

The main reason that I wrote this book is because I couldn't find any other books that covered the content in this format.
I believe that the content is covered with enough detail, but not too much to be overwhelming. This make a great review guide.
This was also written to assist other candidates and help them prepare adequately for their CCIE lab.

I trust you will enjoy reading CCIE R&S Short-Notes and hopefully use it as a reference for years to come.

CONVENTIONS

- CONFIG-SETS - Are short summarized examples showing how to implement various technologies
- COMMANDS - Lists the command syntax, will required and optional strings

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

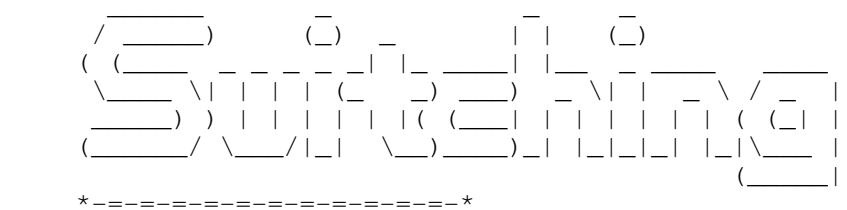
- Command Elements:
  - `|` Vertical bars - Functions as a OR. Line1|Line8
  - `[]` Square brackets - Indicates optional strings
  - `{}` Braces - Indicates required strings

  - `(o)` Optional - Indicates optional, non-required commands

FEEDBACK

By letting me know of any errors and typos, I can correct them for the benefit of future releases.
I would really appreciate it.

If you have questions, comments, or feedback, please feel free to contact me: <notes@ru.co.za>



```
*-=-=-=-=-=-=-=-=-=-=-=-=-*
|      INDEX              |
*-=-=-=-=-=-=-=-=-=-=-=-=-*
- Switchports
      + Speed/Duplex
      + Dynamic
            o Desirable
            o Auto
      + Access
      + Trunk
            o Encapsulation
                  # ISL
                  # 802.1q
            o Mode
                  # Static
                  # DTP
      + Allowed List
      + Tunnel
            o 802.1q Tunnel
- VTP
      + Domains
      + Modes
            o Server
            o Client
            o Transparent
      + Authentication
      + Pruning
            o Prune Eligible List
      + Extended VLANs
- Layer3 Routing
      + Router-on-a-Stick
      + Native Routed Ports
      + SVIs
- EtherChannel
      + Dynamic
            o PAgP
            o LACP
      + Static
      + Layer3 & Layer2
      + Load Balancing
- Spanning-Tree Protocol
      + Root Election
      + Path Selection
            o Port Cost
            o Port Priority
```

```
      + Advanced Spanning-Tree Features
            o Portfast
            o Uplinkfast
            o Backbonefast
            o BPDU Guard
            o BPDU Filter
            o Loopguard
            o UDLD
      + Disabling STP
- Multiple Spanning-Tree Protocol (MSTP)
      + Root Election
      + Path Selection
- Rapid Spanning-Tree Protocol (RSTP)
- Advanced Catalyst Features
      + Flex Links
      + Private VLANs
      + SPAN
      + RSPAN
      + Flow-Control
      + Optimizing System Resources (SDM)
      + Link state Tracking
      + Macros
      + CAM Maintenance
            o Static Entries
            o Aging
            o Logging
            o MAC address notification traps
            o Unicast MAC address filtering
- Bridging
      + Transparent
      + CRB
      + IRB
      + Fall-Back Bridging
            o Aging Time
            o Filtering by Specific MAC Address
            o Adjusting STP Parameters
- Security
      + Port Security
            o Violation
                  # Protect
                  # Restrict
                  # Shutdown
            o MAC Addresses
            o Aging
                  # Time
                  # Type
                  # Errdisable Recovery/Detect
      + 802.1x Authentication
      + Storm Control
      + DHCP Snooping
            o Option-82 Data-Inspection
```

```
      + Ip Source-Guard
      + DAI (Dynamic ARP Inspection)
      + VACLs
            o IP Acl
            o MAC Acls & Ethertypes
      + Port Protection
            o Switchport Protect
            o Switchport Block
- Troubleshooting Switching


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
    Switchports
*======================*
- Speed mismatches usually causes a link to be UP/DOWN.
- Duplex mismatches will bring the link UP/UP but will typically result in packet loss and interface errors
 > Seen with the command "sh interface" as 'late collisions'.

- Layer2 Switchports
 > Access ports
      >> Belong to only one VLAN
 > Trunk ports
      >> Carry multiple VLANs
 > Tunnel interfaces
      >> Transparent layer2 VPN


- Layer3 Routed Ports
 > Switched Virtual Interfaces (SVI)
      >> Logical layer3 VLAN interface.
      >> Configured with "interface vlan{no}"

 > Native routed interfaces
      >> Standard ethernet interfaces where an IP is applied directly to the interface and used for routing.
      >> Configured with "no switchport"

- Trunks
 > ENCAP: ISL
      >> Cisco proprietary.
      >> All traffic is encapsulated within a 30-bytes ISL frame (26-byte header and 4-byte trailer).
      >> Configured with "sw trunk encapsulation isl".
 > ENCAP: 802.1q
      >> Open standard.
      >> All traffic are tagged with 4-byte 802.1q, except the 'native' VLAN.
      >> Supports a native VLAN
            + 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.
            + Default native VLAN is 1.
      >> 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)
      >> Enabled by default
      >> Default mode depends on the platform:
            + 3550 Default mode: Dynamic Desirable (DD) : actively initiates the trunk negotiation.
            + 3560 Default mode: Dynamic Auto (DA) : responds only if trunk negotiation requested.
      >> To negotiate a trunk, at least one side must be DD or be static  'ON'
      >> (DD + DD) = Will trunk. eg ports between 3550 & 3550.
      >> (DD + DA) = Will trunk. eg ports between 3550 & 3560.
      >> (DA + DA) = Will not trunk by default.
      >> DTP negotiation can only be disabled with "sw nonnegotiate".
      >> 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 routers trunk interface.


- Allowed-list
 > Limits which VLANs are allowed on a specific trunk link.
 > aka VLAN minimization. Is when a VLAN is removed from the allowed-list.
 > VLAN-1 is different than other VLANs, in that only data traffic is then not allowed.
      >> Control-plane traffic (CDP,VTP,STP) will still traverse the link using VLAN 1.

- 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.
NOTE: Confirm prior to configuration that underlying end-to-end connectivity is established.
 > When using dot1q tunneling CDP, STP & VTP are NOT carried across the tunnel by default.
 > Additionally dot1q also supports etherchannels 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.

PITFALL: Careful when running OSPF to a switch with a system MTU of 1504, the adjacency won't come up, due to a MTU mismatch.
      Disable the MTU check on the routers OSPF interface with "ip ospf mtu-ignore"


CONFIG-SET: Dot1Q-Tunnel Interface
+-------------------------------------------------
|     system mtu 1504                              STEP1 - Configures the required MTU size (this requires a restart)
|     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, ie the METRO TAG
|      l2protocol-tunnel {cdp | vtp | stp}               - (o) CDP: Re-enables CDP for that interface
|                                                        - (o) VTP/STP: Allows the transport of 3rd party layer2 protocols
|
```

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


#vlan dot1q tag native                                 - Enables native VLAN traffic to get encapsulated with dot1q header
#interface range fa0/13 - 21                           - Configures the range of ports
 #sw mode access                                       - Manually set interface to access mode, disables DTP
 #sw mode trunk                                        - Manually set interface to TRUNK unconditionally (changes mode = on)
 #sw mode dynamic {auto | desirable}                   - {auto}: Will only respond to DTP trunk negotiation requests
                                                       - {desirable}: Will initiate trunk negotiation through DTP
 #sw nonegotiate.                                      - Disables DTP negotiation

 #sw access vlan {vlan}                                - Assign a VLAN to an access port
 #sw trunk encap {isl|dot1q}                           - Manually configure the encapsulation mode. (default = ISL)
 #sw trunk native vlan {vlan id}                       - 802.1q : Changes the (default = 1) native VLAN

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

#system mtu {mtu}}                                     - Configures the required MTU size (this requires a restart)
#system mtu routing {mtu}                              - Sets the MTU for routing processes to a different value than system MTU
#interface fa0/1                                       - Switch interface facing the end point/customer for dot1q-tunnel config
 #sw mode dot1q-tunnel                                 - Enables the dot1q-tunnel on each end-point of the tunnel
 #sw access vlan {vlan id}                             - This is the switch end-to-end VLAN, aka metro-tag
 #l2protocol-tunnel {cdp | vtp | stp}                  - (o) CDP: Re-enables CDP for that interface
                                                       - (o) VTP/STP: Allows the 3rd party to attach his layer2 network directly
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    VTP
*=====================*
- Is not a requirement of ethernet networks, as it does not define broadcast domains.
- Is used to advertise VLAN attributes and ease administration.
- 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
      >> Receives their configuration from the VTP server. VTP changes can't be done on clients.
      >> VLAN configuration is stored in the VLAN database file called vlan.dat and is located on flash (const_nvram).
 > Transparent
      >> Maintains local database, with the VLAN configuration stored in the running config.
      >> Transparent mode is needed to configure 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 does 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 and will not increase with database changes.
 > For every change in Server mode the revision number will be increased by 1, and will be propagated to 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 wiping the existing VTP databases. Regardless if configured as VTP server or VTP client.

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

- VTP Pruning
 > Eliminates the need to statically remove VLANs from trunk links where they 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 what VLANs are allowed to be pruned or not, across a link, based on what VLANs are 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 extenal TFTP server.

-----------
 COMMANDS
-----------
# sh interface [int] pruning                    - Shows pruning status after configuring 'vtp pruning'
# sh interface trunk                            - Shows which local interface are trunked
# 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 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. (must be globally the same)
#vtp pruning                                          - Enables VTP pruning, (must be globally enabled)
#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
*=====================*
```

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

- Troubleshooting trunking and ports
 > When having layer2 issues between routers across multiple switches, an easy way to find the problem:
      >> Create a SVI in that 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://blog.ru.co.za/2008/11/05/troubleshooting-vlan-issues/

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

- Router-on-a-Stick
 > Layer2 switch trunks to external layer3 router.
 > Legacy version of SVI.
 > Routers do not support DTP.
 > Switch interface must be set to a trunk with 'switchport mode trunk'
 > Routers encapsulates ISL or 802.1q traffic using sub-interfaces:

```
      #interface fa0/1.123
       #encapsulation {isl|dot1q} {vlan} [native]
```

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

- Etherchannels are independent of the underlying interface mode, ie 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 etherchannel.
- Important to remember when the command 'channel-group' is issued, the 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 etherchannel 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 Etherchannel
 > Successful layer2 etherchannel will show (SU) with the command "sh etherchannel-channel summary".

CONFIG-SET: Layer2 Etherchannel
+-------------------------------------------------
|     interface range fa0/20-22
|      shut                                       - Shut the physical interfaces before configuring to avoid common issues
|      switchport trunk encapsulation isl
|      switchport mode trunk                      - This would enable layer2 channel on the interfaces
|      channel-group 34 mode desirable            - Specifies channeling protocol: PAgP
|      no-shut
|     !
|     interface port-channel34
|      sw trunk encap isl                         - Configures the layer2 channel parameters
|      sw mode trunk
|

- Layer 3 Etherchannel
 > Shutdown the member interface before configuring the etherchannel.
 > !!! Issue the "no switchport" command on all the member interfaces !!!
 > Successful layer3 etherchannels will show (RU) with the command "sh etherchannel summary".

CONFIG-SET: Layer3 Etherchannel
+-------------------------------------------------
|      interface range fa0/15 - 18
|      shutdown                                   - Shut the physical interfaces before configuring to avoid common issues
|      no switchport                              - This would enable layer3 channel on the interfaces
|      channel-group 12 mode active               - Configures the etherchannel with the channeling protocol: LACP (802.3ad)
|     !
|     interface portchannel 12
|      ip address 10.10.10.1 255.255.255.0        - Configures an IP address on layer3 channel
|     !
|     interface range fa0/15 - 18
|      no shutdown                                - Bring the member interfaces and the portchannel up
```

```
- Etherchannel Load-Balancing options are configured with "port-channel load-balance {mode}":
  dst-ip        Destination IP Address.
  dst-mac       Destination MAC Address(Default for IPv4 and non-IP traffic).
  src-dst-ip    Source XOR Destination IP Address.
  src-dst-mac   Source XOR Destination MAC Address.
  src-ip        Source IP Address (Default for IPv6 traffic).
  src-mac       Source MAC Address.

-----------
 COMMANDS
-----------
# sh etherchannel summary                        - Oneline summary per channel-group, the status of the channel and interfaces
# sh etherchannel load-balance                   - Displays the load-balancing configuration mode
# sh etherchannel {id} port-channel              - Shows port-channel specific information
# sh spanning-tree vlan {vlan id}                - Verifies layer2 channel
                                                 - If one sees member interfaces in FWD mode, then a channel is broken
# sh interfaces trunk                            - Verifies layer2 channel
                                                 - Member interface should not be seen as trunks
# sh ip route                                    - Verifies layer3 channel
                                                 - Should see the portchannel interfaces installed not the member interfaces
# sh lacp sys-id                                 - Verifies 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 etherchannel, specify the channeling protocol
 #channel-protocol {lacp|pagp}                   - (o) Sets the protocol used to manage channeling


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===========================*
    Spanning-Tree Protocol
*===========================*
- Used to prevent layer2 bridging loops.
- PvST is enabled by default.
- PvST is Cisco proprietary.

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

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

- The switch which gets elected root bridge:
 > Will show 'this bridge is root' from "sh span vlan".
 > Will show the same priority and MAC for both 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) based on:
 1st> Lowest cumulative cost to the root:
      >> Inverse value based on interface bandwidth (Iinterface with higher bandwidth will have a lower cost).
 2nd> Lowest upstream BID:
      >> Used to isolate multiple connections to the same upstream bridge.
 3th> Lowest port ID
      >> Lowest port priority (0-255) (default = 128)
      >> Lowest port number ie 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 downstream switches, as cost is the sum of all 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 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 message to other switches.
      >> Max Age (20 sec)           - Age limit when outdated received protocol information is discarded.
      >> Forward Delay (15 sec)     - is 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 are neither root nor a designated port.
 > Alternate port
      - Is a port that is a candidate root port in 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, and 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 now can 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 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 have to be equal values.
 > Blocking state delay ONLY applies when a port first initializes, ie after a reboot, not when a port transitions to forwarding.
 > When a port transitions to forwarding state, the is only listening and forwarding delay.
 > 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.
 > Keep this in mind, on how a question could be asked.

- Portfast
 > Is used to bypass the forwarding delay, thus a port transitions immediately to a forwarding state.
 > Enabling this 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 forwarding state.
 > The CAM table is flooded out of this new root port to expedite the learning phase of upstream neighbors.
 > Configured globally with "spanning-tree uplinkfast"

- Backbonefast
 > Cisco proprietary
 > Used to speed up convergence when a indirect failure occurs upstream in the network by immediately expiring the MAX_AGE timer.
 > Will generate RLQ (Root Link Query) PDU's to check if it should expire max_age for its current BDPU's 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 bring the interface out of err-disable state automatically after 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
 > Drops all inbound BDPU's and does not send BDPU's out of the interface.
 > Unlike BPDU guard, the interface does not go into err-disable state when violation occurs.
 > Other user traffic will still be forwarded.
 > If BPDU filter default is enabled with portfast, all interface will run in portfast mode except those which are receiving BPDU's.
 > Configured globally with "spanning-tree portfast bpdufilter default"
 > Interface configuration "spanning-tree bpdufilter enable"

- ROOT Guard
 > Similar to BDPU guard, but the difference is a root guard interface is only disabled if a superior BPDU is received,
      placing the interface into ROOT_INCONSISTANT_STATE.
 > It should be enabled on a downstream 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 forwarding state erroneously, a loop can occur.
 > Blocked ports will be transitioned into LOOP_INCONSISTANT_STATE to avoid loops.
 > Interface configuration "spanning-tree guard loop"

- UDLD (Unidirection 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"
 > 2 modes:
      >> Normal - informational mode, generates a log entry, but doesn't disable or shutdown the port.
      >> Aggressive - will place a interface into 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 Spanning-Tree
 > STP cannot be disabled directly on a per interface basis.
 > One can turn off Spanning Tree Protocol (STP) on a per-VLAN basis, or globally on the switch.
 > Use the "no spanning-tree vlan vlan-id" command in order to disable STP on a per-VLAN basis.
 > However by filtering BPDU's on a interface one will effectively disable STP running on that interface.
      use the command "spanning-tree bpdufilter enable".
 > FLEX-Links also disables STP on an interface.

```
-----------
 COMMANDS
-----------
# sh spanning-tree summary                              - Shows the STP mode, summary of all vlans timers.
# sh spanning-tree root                                 - Shows 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 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}: Need to be increments of 4096. Lowest numerical value is best
#spanning-tree vlan {id/s} root {primary | secondary}  - {primary}: Configures a priority of 4096
                                                        - {secondary}: Configures a default priority of 28672

#no spanning-tree extend system-id                     - Disables ext-sys-id. (default = enabled) (PVST & Rapid PVST only)
#spanning-tree vlan {id/s} hello-time                  - Sets the hello interval (default = 2sec for RSTP)
#spanning-tree vlan {id/s} forward-time                - Sets the forward delay (default = 15sec)
#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 bdpu filter
#spanning-tree uplinkfast                              - Enables uplinkfast feature
#spanning-tree backbonefast                            - Enables backbonefast feature

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


#interface Fa0/2
 #spanning-tree [vlan] cost {value}                    - Adjusts the path portcost 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}          - Don't send or accept any BPDUs on a interface. Silently discards
 #spanning-tree bpduguard {enable | disable}           - Don't accept BPDUs on this interface, violation = err_disable
 #spanning-tree portfast {enable|disbale} [trunk]      - Enables portfast, and optionally even if in trunk mode
 #spanning-tree guard root                             - Enables STP Root Guard for the interface
 #spanning-tree guard loop                             - Enables STP Loop Guard for the interface
 #spanning-tree guard none                             - Disables the interface guard mode filters
 #spanning-tree link-type {shared | point-to-point}    - Specify a link type for spanning tree protocol use
 #udld port [aggressive]                               - Enables UDLD protocol for copper interfaces, optionally as aggresive

 #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] Will shows more information per interface per VLAN.

#spanning-tree mode mst                               - Configures the spanning-tree mode to MST
#spanning-tree mst configuration                      - Enter MST config sub-mode
 #name MST1                                           - Sets configuration name
 #revision 1                                          - Sets configuration revision number
 #instance 1 vlan 1-200                               - Assign VLANs 1-200 to instance 1
 #instance 2 vlan 201-4094                            - Assign rest of the VLANs to instance 2
#spanning-tree mst 1 priority 0                       - Sets the bridge priority for the spanning tree instance 1 to 0


#interface fa0/4
 #spanning-tree mst {instance} cost {value}           - Change the interface spanning tree path cost for an instance
 #spanning-tree mst {instance} port-priority {value}  - Change 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 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 blocking state will receive STP info, but not send any out that 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.

-=-
================================
    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 silently. If the interface is down, traffic to that MAC will 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.

- 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.

- RSPAN
 > 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 port where 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 which 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.
 > Flow-control is a older technology to control the sending rate of a host, newer MLS QOS technologies are more evolved.
 > It is recommended to turn off 802.3X flow control when MLS QoS is enabled.
 > Catalyst 3560 ports can receive, but not send, pause frames.
 > By default flowcontrol 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.

- Voice VLAN (VVLAN)
 > Most Cisco phones have a built-in 3-port switch and is able to distinguish the phone and the PC using different VLANs and optionally 802.1p COS.
 > Voice config is communicated via CDP to the IP phone.
 > 3 different connecting options:
     1. Separate DATA VLAN / VOICE VLAN.
           >> VOIP frames are tagged with COS 5.
           >> Connection between switch and IP phone is a 802.1q trunk with native VLAN equal to data VLAN.
           >> Configured with "switchport voice vlan" command.
     2. Single VLAN for both VOICE and DATA
           >> Frames are not tagged, thus the phone merely acts as a switch.
           >> Connection between switch and IP phone is configured as a ACCESS link.
           >> If "no switchport voice vlan" configured, then option 2 automatically applies.
     3. Single VLAN for DATA and VOICE but with COS 5 marking
           >> DATA traffic is marked as COS 0 within a 802.1q header.
           >> VOICE traffic is marked as COS 5 within the 802.1q header.
           >> COS zero will be accepted as the access VLAN.

- Link-State Tracking
 > Link-state tracking, also known as trunk failover, is a feature that binds the link state of multiple interfaces.
 > Its 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, that 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 (ie 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.

- Flex Links
 > Used as an alternative to STP in environments where physical loops occur in the layer2 network.
 > Works similar to "backup interface", whereby one has 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, that are defined by primary VLAN and secondary VLANs.
 > Basically it is 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!!!
 > Difference between PVLAN and protected port, PVLAN can span multiple switches whereas protected ports don'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 ports only 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 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

# debug back all                                      - Enables debugging for the backup interface

#mac-address-table static {mac} vlan {id} int         - Hardcode a MAC address to a interface
#mac-address-table static {mac} vlan {id} drop        - Null-switch a MAC address

#monitor session 1 source {int | vlan}                - Specify the local source interface of the traffic to span
#monitor session 1 dest int {int} [encap | ingress]   - Setup SPAN to 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]- Specify the source of the traffic to span and the direction (Def=BOTH)
#monitor session 1 destination remote vlan 200       - Fa0/2 received traffic is redirected to the RSPAN VLAN-200
#monitor session 1 source remote vlan 200            - Configures another switch to receive 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 requires 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 or 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}                                      >>> Creates custom macro to configure multiple interface <<<
 switchport mode access                                 - By using a #, the line will act as description
 switchport access vlan 146
 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}
                                                        - Alters 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 or 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 PRI VLAN 1 and SEC VLAN 10. Can talk to any ports in 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:
 > Router has IP routed.
 > Switches has 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 the broadcast domain for one protocol needs to extended while maintaining it for another protocol.
!!NOTE!! Routers running in bridged mode doesn't support the sys-id-ext, so the bridge priority will be 32768 only, for any VLAN.
This would make the router 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.

 > Steps to configure
 1. Create transparent bridge group
 #bridge {num} protocol ieee
 2. Enable IRB and what to be bridged
 #bridge irb
 #bridge {num} route {protocol} OR - Enables routing and bridging
 #bridge {num} bridge {protocol} - On by default
 3. Enable routing & bridging for the bridge-group under the interface
 #interface fa0/0
 #bridge-group {num}
 4. Configure BVI to connect the bridged and routed domain
 #interface BVI{num}
 #ip add 1.1.1.1 255.255.255.0

- 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.
 > Steps to configure
 1. Specify the bridging VLAN
 #bridge 1 protocol vlan-bridge
 2. Assign the SVI and routed port to this bridge.
 #interface vlan1
 #bridge-group 1
 #interface fa0/1
 #no switchport
 #bridge-group 1
 3. Verify config:
 # sh bridge 1 group

```
-----------
 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 go on the BVI

#bridge 2 protocol vlan-bridge                           - Enables fallback bridge group
#interface vlan 2
 #bridge-group 2                                         - Applies 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 a etherchannel nor be a private-VLAN port.
      Can be configured, but won't work.
 > NOTE that when using HSRP etc, to also allow HSRP's MAC address on a port.
 > Occasionally when port-security is configured with 2 secure-MAC addresses, the port might still go err-disable on two
      MAC addresses. Try to increase allowed amount to three.
 > Violation mode
      + Shutdown
            o Default mode
            o Upon violation the port changes to err-disable state.
            o Generate SNMP/Syslog.
      + Protect
            o Violators cannot send traffic in.
            o This mode disables learning when any VLAN reaches the max limit, not recommended on trunk ports.
      + Restrict
            o Violators cannot send traffic in.
            o Generates SNMP/Syslogs.
```

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

- Storm Control
 > Limit the amount unicast/broadcast/multicast traffic accepted on a port.
 > Traffic above multicast rate suppresses unicast, broadcast and multicast.
 > With storm control it recommended to hardcode the interface speed to get around 10/100/1000 negotiation issue.
 > Configured with "storm-control {broad | multi | unicast}"

- 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 Insertion
 >> 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 switch

```
|Configured on SW1 that is connected to VLAN-17 where the DHCP server (R1) is connected
|
|     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
|     !                                             - ie option-51 (Refer to IP-Serv 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, including access and trunk ports.
 > One can configure IP source guard with source IP address filtering or with source IP and MAC address filtering.
 > Requires DHCP snooping to be enabled, else 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
 > Is 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 a ACL-permit performs the "forward"/"drop" function in the access-map. A ACL-deny will be ignored.
    So to deny traffic with VLAN ACL's, permit the traffic and use a "drop" action in the access-map.

 > MAC-ACL's will only match NON-IP traffic.
 > 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 memorise!
    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 echo's & IPv6 on VLAN-162 but forward all other

```
|     access-list 101 permit icmp any any echo        - Matches IP ICMP echo
|     !
|     mac access-list extended EtherType
|      permit any any 0xAAAA 0x0                      - Matches specific ethertype (STP, VTP, PAgP, PVST, DTP, CDP, UDLD)
|     !
|     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 ethertype for IPv6
|     vlan access-map VACL 30
|      action forward                                 - Forwards all other traffic
|     !
|     vlan filter VACL vlan-list 162                  - Applies access-map
```

- Port Protection
 > Difference between PVLAN and protected port, PVLAN can span multiple switches whereas protected ports doesn'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 is 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 behaviour between a protected port and a non-protected port is as usual.
 > If configured on an etherchannel, it applies to all ports in the group.
 > Configured with "switchport protected".

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



-----------
 COMMANDS
-----------
# sh port-security                                    - Shows the counters per secure-port, ie MAC, violation count, status
# sh port-security {interface}                        - Shows more verbose output about the interface specified
# sh dot1x                                            - Verifies dot1x configurations
# sh storm-control                                    - Shows storm-control specifics
# sh ip dhcp snooping                                 - Displays the DHCP snooping configuration for a switch
# sh ip source binding                                - Displays the IP source bindings on a switch
# sh ip verify source                                 - Displays the IP source guard configuration on the switch


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


                                                     >>> Errdisable Recovery <<<
#errdisable recovery psecure-violation               - Example enable 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 1 or all applications


                                                     >>> 802.1x Authentication <<<
#aaa new-model                                       - (R) Enable aaa
 #aaa authentication login default none              - (R) Disables AAA for all other authentication methods
 #aaa authentication dot1x [default group radius]    - (R) Create 802.1x authentication method list querying a radius server
#dot1x system-auth-control                           - (R) Enable 802.1x authentication globally on the switch
#interface fa0/3
 #dot1x port-control auto                            - (R) Enable 802.1x authentication for the port
#ip radius source-interface loopback0                - (o) Optionally 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, eg when ports are 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}                   - (R) Creates the access-map for VLAN-ACL
 #match mac address {acl}                       - (R) Used to match MAC-address or
 #match ip address {acl}                        - (R) Used to match ACL entries
 #action {drop|forward}                         - (R) Action that is applied to the match
#vlan filter {name} vlan-list {all | (vlan-id)} - (R) Applying the VLAN-ACL


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

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
   Troubleshooting Switching             >>>   {} curl-brackets indicates replaceble values      <<<
*==============================*
*******************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*******************************************************************************************************************************


- When troubleshooting interfaces and trunks, consider the following:
 > Confirm the state of the interfaces                                                    # sh int | i line
      >> If a interface is UP/DOWN, is it caused by a speed mismatch?                     # sh int status
      >> Is there a duplex mismath?                                                       # 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 sucessfully?            # 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 a 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 VLAN's 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 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://bit.ly/ruhann-ts-vlan


- 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 a server's!      # sh vtp status | i Revision
      >> If not change it to zero, by changing mode to tranparent 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 etherchannels, consider the following:
 > What are the state of the ports and the channel status?                                # sh etherchannel 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 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 BDPU's filtered potentially causing STP loops?                                 # sh run | i bpdufilter|backup int
 > Is spanning tree disabled for a specific vlan?                                         # sh spanning-tree vlan 20
 > Are any interfaces in ERR-DISABLE state?                                               # sh int status
 > Are error recovery enabled for the required services?                                 # sh errdisable recovery
```

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

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


-=-

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

Frame-Relay Operation

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

- DOC-CD LOCATION
  - > 12.4T Configuration Guides
    - > WAN
      - > 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 wide area network (WAN).
- The logical communication path between two or more DTEs (routers) are called VCs (virtual circuits).
- VCs (Virtual circuits) may be permanent (PVCs) or switched (SVCs). PVC'sare more common.

- DLCI (DataLink Connection Identifiers)
 > DLCI's are used as a frame-relay address, which identifies the VC over which frames should travel in a frame-relay cloud.
 > It is contained within a 10-bit field inside the frame-relay header.
 > DLCI's are locally significant to a link and can change as it passes through the network.
 > To see active DLCI's issue the command "sh frame-relay map".
 > To see all the DLCI's 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 (router) about the DLCIs and status of each VC.
 > These inquiry/status messages functions as, and are referred to as LMI keepalives too.
 > LMI can enabled/disabled by using the keepalive/no keepalive commands.
 > LMI holdtime is 3x keepalives. LMI holtime cannot be adjusted directly, but only by changing the keepalive interval times three.
 > If 3 keepalives (default) are missed an interface will be considered down.
 > There are three LMI types: Cisco/ANSI/q933a.
 > LMI autosense is enabled by default, which 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 what DLCI is associated with what interfaces/sub-interface.
 > The command "encapsulation frame-relay" enables LMI automatically.

- LMI Keepalives and Full Status Update
 > By default, LMI keepalives are sent every 10 seconds.
 > Keepalives must match, to prevent flapping interfaces.
 > If LMI autosense is unsuccessful, an intelligent retry scheme is built in.
 > Every N391 interval (default is 60 seconds, which is 6 keepalives at 10 seconds each), LMI autosense will attempt to ascertain the LMI type and request a complete status info about each VC. This is also known as full status update.
 > If required to change the full status update timers, 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, (180sec / 10 sec keepalive = 18), thus 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 frame-relay switches to deliver frames across the frame-relay network. 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. It also includes a Network Layer Protocol ID or NLPID field is commonly used to indicate information about the data-link layers.
 > The frame-relay frames 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: Encapsulations per-interface and per-DLCI examples
+------------------------------------------------------------
|     interface s1/0
|      encapsulation frame-relay ietf                       - Sets IETF encapsulation as default at the interface level
|      frame-relay map ip 131.108.123.2 48 broadcast        - Here the default encapsulation method for all maps default to IETF
|      frame-relay map ip 131.108.123.3 49 broadcast cisco  - Per-DLCI encapsulation overwrites per-interface encapsulation
|     !
|     interface s1/1
|      encapsulation frame-relay                            - Default interface encapsulation is Cisco
|      frame-relay map ip 131.108.143.2 58 broadcast ietf   - Per-DLCI encapsulation overwrites per-interface encapsulation
|      frame-relay map ip 131.108.143.3 59 broadcast        - Here the default encapsulation method for all maps default to 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 or switch noticing the FECN, will set the BECN bit on traffic returning to the source, to indicate congestion and
      possible instruct the source to slow down transmission.
 > The DE (Discard Eligibility) is used to indicate traffic that are in violation of the conformed rate, might be subject
      to discarding during periods of congestion. Frames marked with DE bit will be dropped before non-marked frames.
 > Refer to 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 DLCI from the frame-switch, the other side is down.
 > Deleted        - Indicates a local config problem. The frame-switch has no such mapping and responded with a "deleted message".
 > Static         - Indicates that LMI was turned off with the "no keepalives".

- 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, you can create a special broadcast queue 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]                             - Displays PVC status, DLCI's, in/output packets, PVC uptime, etc

# debug frame-relay packet                              - Shows the DLCI mappings
                                                        - Should actually be 'debug fr frame', not packet :)
                                                        - 'encaps failed- no map entry" shows incorrect DLCI assignment
#interface s0/1
 #encapsulation frame-relay [ietf]                      - Enables frame-relay encapsulation on a 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 an interface.
 #cdp enable                                            - Enables CDP on an interface
```

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

- Broadcast Replication
 > Frame-relay does not have the capability to send a single frame over multiple PVC's to multiple destinations.
 > But the broadcast functionality is still sometimes required by routing protocols.
 > Also known 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 the 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 (via InARP).

- InARP (Inverse ARP)
 > Is used to dynamically resolve the 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 a 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 is needed.
 > When a point-to-point interface is connected to a 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 a 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
 #encap 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 map a remote IP address to a local DLCI
                                                       - [broadcast] Enables frame-relay broadcast relay across the PVC
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===================================*
   Interface Types
*===================================*
- 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 PVC's(layer2 circuits).
 > Requires layer3-to-layer2 resolution through either InARP or manual mapping. (Refer to previous section)
 > 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 frame-relay links
 > Are router-to-router serial links running frame-relay encapsulation, but with no frame-relay switch in between 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 s0/0                                      >>> 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 broadcasting for this host.
|     !
|     interface s1/1
|      encapsulation frame-relay
|     !
|     interface s1/1.1 point-to-point                     >>> Point-to-Point interface <<<
|      ip address 10.0.1.4 255.255.255.0
|      frame-relay interface-dlci 104                     - Assigns this interface the DLCI-104 (only one PVC)
|     !
|     interface s1/1.2 multipoint                         >>> Multipoint interface <<<
|      ip address 10.0.2.4 255.255.255.0                  - Here the interface will rely on dynamic InARP mappings received.
|     !
```

```
|     interface s1/5                                    >>> Back-to-Back interface <<<
|      ip address 10.1.5.1 255.255.255.0
|      encapsulation frame-relay                        - Enables Cisco encapsulation 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 192.43.96.9 255.255.255.0             - Assigns the logical interface an IP address
|      frame-relay interface-dlci 789                   - Assigns the PVC identifier
|      multilink bandwidth-class b                      - Both links must be up before the bundle is brought up
|     !
|     interface Serial0/2/1
|      no ip address
|      encapsulation frame-relay mfr1                   - Assigns the first interface to the bundle
|     !
|     interface Serial0/2/0
|      no ip address
|      encapsulation frame-relay mfr1                   - Assigns the second interface to the bundle
|
```

- Interface states
 > The physical interface connecting 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 DLCI's defined, all DLCI's must be down before the interface will show down/down.
    If one DLCI is up, the interface will be up/up.
 > http://blog.ru.co.za/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.

!TIP! 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 it.

- 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 local IP on frame-relay interface

```
|     interface Serial0/1/0
|      ip address 191.1.34.3 255.255.255.0            - Configures the interface IP
|      encapsulation frame-relay
|      frame-relay map ip 191.1.34.4 304 broadcast    - Maps the remote-end IP to local-DLCI
|      frame-relay map ip 191.1.34.3 304              - Maps the local IP to local-DLCI, thus enabling the recursive mapping
|     end                                                   for remote-end router to redirect packets back
|
```

COMMANDS

```
# sh frame-relay map                                  - Shows the DLCI mappings, status, dynamic/static, LMI types
# sh frame-relay multilink                            - Displays the current frame-relay multilink configuration
# sh interfaces mfr {mfr-interface}                   - Displays information and packet statistics for the bundle interface

#interface s0/1
 #encapsulation 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 layer2 circuit terminates on the interface
 #frame-relay map {prot}{ip}{dlci}[broadcast]         - Statically map a remote IP address to a local DLCI
                                                      - Broadcast must be manually enabled

#interface s2/1
 #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
*====================================*
- 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.
- Example:
      R1----R2----R3


- Four possible solutions:
 > Add additional static mappings via the hub router.
 > Change to point-to-point sub-interfaces.
 > Use static IP routing with next-hop instead of interface.
 > Use layer3 dynamic routing, like OSPF interface type point-to-multipoint.



CONFIG-SET: Hub-and-Spoke example with static mappings, R2 as hub and R1, R3 as spokes
+------------------------------------------------------------------
|R2>>
|     interface s2/0                                    - R2 is the hub
|      encapsulation frame-relay
|      ip add 192.168.0.2 255.255.255.0
|      frame-relay map ip 192.168.0.1 201 broadcast          - Static mapping to each spoke allowing broadcast replication
|      frame-relay map ip 192.168.0.3 203 broadcast          - Static mapping to each spoke allowing broadcast replication
|
|R1>>
|     interface s1/2                                    - R1 is a spoke
|      encapsulation frame-relay
|      ip add 192.168.0.1 255.255.255.0
|      frame-relay map ip 192.168.0.2 102 broadcast          - Static mapping to the hub
|      frame-relay map ip 192.168.0.3 102                    - Static mapping to other via the hub
|
|R3>>
|     interface s0/2                                    - R3 is a spoke
|      encapsulation frame-relay
|      ip add 192.168.0.3 255.255.255.0
|      frame-relay map ip 192.168.0.2 302 broadcast          - Static mapping to the hub
|      frame-relay map ip 192.168.0.1 302                    - Static mapping to other via the hub
|
```

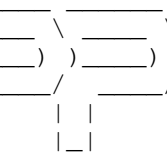
```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
   Bridging Frame-Relay Links
*====================================*
- The frame-relay bridging software uses the same spanning-tree algorithm as the other bridging functions.
- The bridging spanning tree views each PVC as a separate bridge port.
- A frame arriving on one PVC can 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.
|
|     bridge irb
|     bridge 1 protocol ieee
|     !
|     interface serial 0
|      encapsulation frame-relay                         - Enables frame-relay transparent bridging
|     !
|     interface serial 0.1
|      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 serial 0.2
|      bridge-group 1                                    - Associates the sub-interface with a bridge group 1
|      frame-relay map bridge 64 broadcast               - Bridges DLCI 42 and 64 together
|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
   Frame-Relay Auto-Install
*====================================*
- Pre-configured frame-relay clients requesting an address via BOOTP can be done using the config-set below.


CONFIG-SET: Frame-Relay Auto-Install
+--------------------------------------------
|     #interface s0/1.1 point-to-point
|      #ip address 192.168.1.1 255.255.255.0
|      #frame-relay interface-dlci 105 protocol ip 192.168.1.5    - This IP will be assigned to the connecting host via BOOTP
|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================================*
 Frame-Relay End-to-End Keepalives (FREEK)
*============================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > WAN
         > Cisco IOS Wide-Area Networking Configuration Guide, Release 12.4T
          > Part 1: Frame-Relay
           > Configuring frame-relay
            > Configuring Frame Relay End-to-End Keepalives

- Adds the ability to track status between DTE devices.
- Freek can be configured on a physical interface, but when the freek status goes down, freek will not bring down the physical
      interface, because it will not know when to bring it back up.
- For this reason it is recommended to configure freek on a sub-interface.
- 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 the same.
      >> 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
-----------
#show frame-relay pvc                                      - Shows the FREEK status as EEK UP or EEK DOWN

#map-class frame-relay FREEK                               - Creates map-class
  #frame-relay end-to-end keepalive mode {bidirectional | request | reply | passive-reply}
                                                           - Enables freek for the class
#interface s1/0.345 {point-to-point|multipoint}
#frame-relay class FREEK                                   - Applies the map-class for the EACH DLCI on the interface, OR
 #frame-relay interface-dlci 402
      #class FREEK                                         - Applies the map-class ONLY for DLCI 402
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
   Troubleshooting Frame-Relay                  >>>   {} curl-brackets indicates replaceble values      <<<
*====================================*
*******************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*******************************************************************************************************************************

- 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 DLCI's 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 shows 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 no valid DLCI's           # 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 DLCI's 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 fr-switch, other side is down.
 > DELETED  - Local config problem. Fr-switch has no such mapping, responds with 'deleted'.
 > STATIC   - Is when 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
 > Has keepalives been disabled? (Here it is required)                                      # sh run | i interface|no keepalive
 > Are both sides using the same DLCI's? (Required for back-to-back)                        # sh frame pvc | i DLCI

- When troubleshooting frame-relay mappings, consider the following:
 > For successful mappings, the PVC's should be in ACTIVE state                             # sh frame pvc | i DLCI
 > To see active DLCI's and their mappings issue the command                                # sh frame map
 > If a sub-interfaces was 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 mappings 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 mapping error.                              #debug frame packet

 > A typical issue with partial frame-relay networks is mapping:
      >> 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 |
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

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

- During IPCP negotiation, if the peer presents a peer IP address during IPCP address negotiation and the peer router has no
      address is assigned, the presented address is acknowledged and used in the current session.


CONFIG-SET: PPP IPCP address allocation
+-----------------------------------------------
|R5:
|     interface Serial0/2/0
|      ip address 192.168.205.5 255.255.255.0
|      encapsulation ppp
|      peer default ip address 192.168.205.2                - Specifies the IP adress to be issued to R2
|      ppp lcp predictive                                   - Specifies link control to predict peer responses
|      ppp ipcp predictive                                  - Specifies IP control to predict peer responses
|
|R2:
|     interface Serial0/1/0
|      ip address negotiated                                - R2 will request a IP address from the other side of the link
|      encapsulation ppp
|      clock rate 2000000
|      ppp lcp predictive
|      ppp ipcp predictive
|
|
>     R2#sh ip int brief | i Serial0/2/0
>     Serial0/2/0     192.168.205.2   YES IPCP   up  up     - Notice the method is IPCP (IP Control Protocol)


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  PPP Authentication
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Dial and Access
         > 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
```

- PPP authentication is a two-way process. A router is a server, a client, or both.

- PAP (Password Authentication Protocol)
 > PAP sends clear text username and clear text password.
 > PAP supports unidirectional (one-way) and bi-directional (two-way) authentication.
 > With unidirectional authentication, only the side receiving the call (server) authenticates the remote side (client).
 > The remote client does not authenticate the server.
 > With bi-directional authentication, each side independently sends an Authenticate-Request (AUTH-REQ).
 > And receives either an Authenticate-Acknowledge (AUTH-ACK) or Authenticate-Not Acknowledged (AUTH-NAK).
 > 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 authentication R2
|
|R1#
|     username R2C password cisco                    - This is the password R2(client) will use to connect to R1
|     !
|     interface s1/0
|      encapsulation ppp
|      ppp pap authentication pap                    - Enables server side
|      ppp max-bad-auth 3                            - A maximum of 3 bad authentication tries allowed
|     !
|
|R2#
|     interface s2/1
|      ppp pap sent-username R2C password cisco      - Enables R2 as the client
|      ppp max-bad-auth 3                            - A maximum of 3 bad authentication tries allowed
|
```

- CHAP (Challenge Handshake Authentication Protocol)
 > Sends clear text username and MD5 password.
 > By default, the router uses its hostname to identify itself to the peer, but can be changed with "ppp chap hostname".
 > A interface level CHAP hostname overwrites the routers global hostname.
 > If the same host name is specified on both sides, the session authentication will fail, as the router ignores a
     authentication-request from its own hostname. To get around that issue the hidden command "no ppp chap ignoreus".
 > A global password is always tried first and then a interface-level password will be tried.
 > CHAP is defined as a one-way authentication method, but if applied in both directions it create two-way authentication.

```
CONFIG-SET : PPP two-way CHAP authentication
+------------------------------------------------------------
|     R2#
|     username CCIE password 0 cisco                  - The UID(CCIE) is the hostname of the peer
|     !
|     interface Serial0/2
|      ip address 10.0.24.2 255.255.255.0
|      encapsulation ppp                              - Enables PPP encapsulation
|      ppp authentication chap                        - Enable the use of CHAP authentication
|
|
|     R4#
|     username R2 password 0 cisco                    - The UID(R2) is the hostname of the peer
|     !                                               - The passwords must match between peers
|     interface Serial1/0
|      ip address 10.0.24.4 255.255.255.0
|      encapsulation ppp
|      ppp authentication chap                        - Enables the use of CHAP authentication
|      ppp chap hostname CCIE                         - Interface hostname overwrites routers hostname
|

- If a ? (question-mark) is required in the password, use CRTL-V or ESC-Q to enter a '?' on the CLI.


-----------
 COMMANDS
-----------
#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 {name} password {pwd}                       - Verifies that an incoming username and password are valid
#interface s0/0
 #ppp authentication pap                              - Authentication response from the server side
 #ppp pap refuse                                      - Disables a client responding to pap. Router is a client by default
 #ppp pap sent-username {USER} 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 {name} password {pwd}                       - Username specified here needs to match remote side hostname
#interface s0/0
 #ppp authentication chap                             - Authentication request from the server side
 #ppp chap hostname {name}                            - Allows alternate CHAP hostname, instead of routers hostname
 #ppp chap password {pwd}                             - Defines a 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
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  Peer Neighbor Route
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Dial and Access
         > 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

- A /32 is automatically created for neighbor routes by default.
- It automatically sets up a route to the peer address on a point-to-point interface when the PPP IPCP negotiation is completed.
- It is useful to provide reachability when both ends of the PPP link are not on the same logical subnet. ie IP-unnumbered.
- Can be safely disabled when both ends of the link are in the same logical subnet.
- If using IP-unnumbered or dissimilar IP subnets on a data-link, leave it enabled.

- Most commonly used when different IP subnets on the same physical segment.
 > R1 S1/0 is directly connected to R2 S2/0.
 > R1 S1/0: IP address is 4.4.4.4/24.
 > R2 S2/0: IP address is 5.5.5.5/24.
 > R2 will have 4.4.4.4 in the routing table as 4.0.0.0/8 and 4.4.4.4 will be pingable.
 > R1 will have 5.5.5.5 in the routing table as 5.0.0.0/8.
 > Problems is that only the IP address is advertised to a PPP neighbor, and not the SUBNET MASK, it will assume classful boundary.

-----------
 COMMANDS
-----------
#interface E0
 #no peer neighbor-route                              - Disables peer neighbor route, then connected IP address won't be advertised
                                                      - Disables /32 host routes for the interface


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

-----------
 COMMANDS
-----------
# show 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
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*================================*
  Link Quality Monitoring (LQM)
*================================*
- The PPP suite includes a feature that allows devices 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


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
  Multilink PPP (MLP)
*========================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Dial and Access
         > 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 add
|      encapsulation ppp
|      ppp multilink                                    - Enables MLP on S0/0
|      multilink-group 2                                - Assigns the interface to multilink group 2
|     !
|     interface s0/1
|      no ip add
|      encapsulation ppp
|      ppp multilink                                    - Enables MLP on S0/1
|      multilink-group 2                                - Assigns the interface to multilink group 2
|     !
```

```
|     interface multilink2                          - Multilink interface is where logical options are configured
|      ip address 192.168.0.1 255.255.255.0
|      ppp multilink
|      multilink-group 2
|
```

- LFI (Link Fragmentation and Interleaving)
 > Interleaving on MLP allows large packets to be multilink encapsulated and fragmented into a small enough size to satisfy the delay requirements of real-time traffic; small real-time packets are not multilink encapsulated and are transmitted between 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.
 > Maximum fragment delay: If one specifies 20 ms delay, MLP will choose a fragment size based on the configured value.

- MCMP (Multi-class 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 in between the fragments of larger data packets.

- MRRU Negotiation
 > DOC-CD LOCATION
  > 12.4T Configuration Guides
   > Dial and Access
    > 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}                - (o) Configure 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
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
  PPP over Frame-Relay (PPPoFR)
*==================================*
 > 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.
- By implementing PPPoFR, authentication of frame-relay PVCs can be implement, or multiple PVCs could be binded together using MLP.
- PPPoFR is configured through the use of a virtual-template interfaces.
- A virtual-template is a PPP encapsulated interface that is designed 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 can affect certain network designs such as using the backup interface command
      on a virtual-template.



CONFIG-SET: PPP over frame-relay (PPPoFR) example
+----------------------------------------------------
|     interface virtual-template1                      - STEP1: Create the virtual-template interface
|      ip address 192.1.7.6 255.255.255.0              - Configures all the logical options like a IP address
|      ppp chap hostname ROUTER6                       - NOTE: "encapsulation ppp" not needed as virtual-templates always runs PPP
|      ppp chap password 0 CISCO                       - Authentication is optional
|     !
|     interface Serial0/0                              - STEP2: Configure the physical frame-relay interface and
|      encapsulation frame-relay                             bind the virtual-template to the frame-relay PVC
|      frame interface-dlci 201 ppp virtual-template1 - Note that the order that these steps are performed are important
|     !
|     !
>     #sh ip interface brief | include 192.1.7.6
>      virtual-access1 192.1.7.6 YES TFTP up up
>      virtual-template1 192.1.7.6 YES manual down down


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
  PPP over Ethernet (PPPoE)
*==================================*
- 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
```

- The PPPoE client feature provides PPPoE client support on ATM PVCs (permanent virtual circuits)and ethernet interfaces.
- A dialer interface must be used for cloning virtual access.
- A PPPoE session is initiated by the PPPoE client.
- PPPoE is a commonly used application in the deployment of digital subscriber lines (DSL).
- The PPP over ethernet client feature expands PPPoE functionality by providing support for PPPoE on the client
  as well as on the server.

- PPP is configured on the physical interface and IP on the logical interface.
- A virtual-template interface is a PPP interface, no need for 'encapsulation ppp'.
- This can be seen with "sh interface virtual-template1"
- It is recommended that you set the MTU to 1492 bytes. This value accommodates a PPPoE header encapsulation of 8 bytes
  in the ethernet frame payload.

- Dialer persistent feature:
 > Allows a dial-on-demand routing (DDR) 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.
 > The dialer persistent command starts a timer when the dialer interface starts up and starts the connection
  when the timer expires.

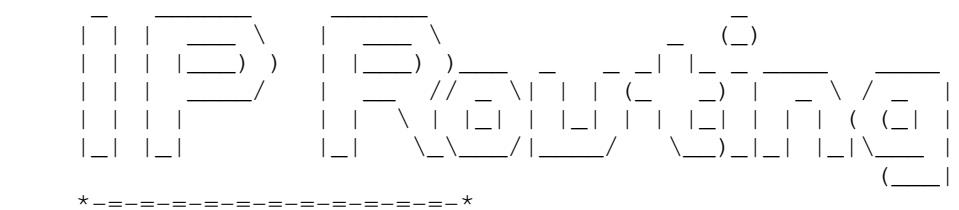
CONFIG-SET: PPP over Ethenet (PPPoE) configuration

```
| R2 > PPPoE SERVER CONFIG
|      !
|      username test password test                      - Defines the username and password to be used by a PPPoE client
|      !
|      bba-group pppoe global                           - Creates the PPPoE group
|        vitual-template 1                              - A virtual-template is used for incoming PPPoE requests instead of a dialer
|      !
|      ip local pool PPPoE 172.16.0.20 172.16.0.29      - IP pool used to allocate IP's to remote PPPoE clients
|      !
|      interface e0/1
|       ip address 10.0.0.6 255.255.255.0
|       pppoe enable group global                       - Enables PPPoE on the physical interface
|      !
|      interface virtual-template1
|       ip address 172.16.0.12 255.255.255.255         - This will be used as the source IP address on virtual-access interfaces
|       ip mtu 1492                                     - Required to cater for PPPoE frame overhead
|       peer default ip address pool PPPoE              - Assigns the pool to be used for assigning IP addresses to the interface
|       ppp authentication chap callin                  - Incoming calls will use CHAP authentication
|
|
|
| R1 > PPPoE CLIENT CONFIG
|      !
|      bba-group pppoe global                           - Creates the PPPoE group
|      !
|      interface fa0/0                                  - Configures the inside interface
|      ip address 10.0.0.5 255.255.255.0
|      !
```

```
|     interface fa1/0                                  - Configures the outside interface
|      no ip address
|      pppoe enable group global                       - Enables PPPoE in group global on the outside interface.
|      pppoe-client dial-pool-number 1                 - Adds the interface into a dialer pool 1
|     !
|     interface dialer 0                               - Configures the PPPoE interface
|      ip address negotiated                           - IP address will be negotiated with the PPPoE server
|      ip mtu 1492                                     - MTU is reduced to cater for PPPoE framing overhead
|      encapsulation ppp                               - Non PPP interfaces need it enabled
|      dialer-pool 1                                   - Associates the dialer interface with a dial pool
|      dialer-group 1                                  - Associate the dialer interface with a dial-on-demand group
|      ppp authentication chap callout optional        - Enable CHAP authentication, only on outgoing calls
|                                                      - (optional) The remote end does not need to authenticate
|      ppp chap hostname test                          - Uses CHAP authentication with username=test
|      ppp chap password 0 test                        - Uses CHAP authentication with password=test
|     !
|     ip route 0.0.0.0 0.0.0.0 Dialer0                - Static default route points to the PPPoE interface
|     dialer-list 1 protocol ip permit                 - Defines the dialer list object matching interesting traffic


-----------
 COMMANDS
-----------
# show vpdn                                           - Displays information about active Layer2 Forwarding (L2F) protocol
# show vpdn session packet                            - Displays PPPoE session statistics
# show vpdn session all                               - Displays PPPoE session information for each session ID
# show vpdn tunnel                                    - Displays PPPoE session count for the tunnel

# 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 {dialer-group} {prot} list {ACL}         - References ACL listing interesting traffic or use the persistent command
#bba-group pppoe {name}                               - Create a PPPoE profile

#interface E0/1
 #pppoe-client dial-pool-number 1                     - Specifies the dialer interface to use for cloning

#interface dialer0
 #mtu 1492                                           - Adjusts MTU size to accommodate PPPoE header encap size of 8 bytes
 #ip address negotiated                              - Specifies the IP to be obtained from the PPPoE server
 #dialer pool {number}                               - Associates the dialer interface with a dial pool
 #dialer-group {number}                              - Configures an interface to belong to a specific dialer group
 #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            >>>   {} curl-brackets indicates replaceble values      <<<
*========================*
*******************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*******************************************************************************************************************************


- 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 and NCP phases as OPEN?                       # sh int {int} | i LCP|NCP
 > For LCP phase there is not much more that can cause problems.                              # debug ppp negotiation
    >> But more info can be seen with a debug. Full details here (http://bit.ly/ppp-nego)

- When troubleshooting PPP authentication, consider the following:
 > PPP authentication does not begin until the LCP phase is complete and is in a OPEN state.    # sh int {int} | i LCP
 > PPP authentication issues are almost always configuration errors!
 > If two-way authentication is required, first get one-way authentication working!
```

```
 > For PAP authentication issues:
      >> Confirm PAP authentication is configured correctly. (server must auth client)          # sh run | i pap|username
      >> Is the PAP server configured as a server?                                              # sh run | i auth.*pap
      >> Do the usernames and passwords match between peers?                                    # sh run | i username
      >> Else use the debug to analyze what the cause of failure is. (AUTH-NAK are bad)         # 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
      >> Is the local routers hostname matching the peers username command?                     # sh run | i username
            >>> If needed, are the neighbors allowed to use the same hostname?                  # sh run | i ignoreus
      >> Else use the debug to analyze what the cause of failure is.                            # debug ppp authentication


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
   OUTPUT-101
*========================*
---->
Example debug output of a successful PPP one-way PAP authentication
#debug ppp authentication
                                                                   >>> I = Incoming, and O = Outgoing <<<
*Mar  1 03:14:37.827: Se2 PAP: O AUTH-REQ id 7 len 18 from"cisco" <- Client sends its username/password to the server.
*Mar  1 03:14:37.831: Se2 PAP: Authenticating peer cisco          <- Performs a lookup for username cisco' and password.
*Mar  1 03:14:37.839: Se2 PPP: Sent PAP LOGIN Request
*Mar  1 03:14:37.847: Se2 PPP: Received LOGIN Response PASS       <- Username/password match is succesful.
*Mar  1 03:14:37.851: Se2 PPP: Sent LCP AUTHOR Request
*Mar  1 03:14:37.855: Se2 PPP: Sent IPCP AUTHOR Request
*Mar  1 03:14:37.863: Se2 LCP: Received AAA AUTHOR Response PASS
*Mar  1 03:14:37.867: Se2 IPCP: Received AAA AUTHOR Response PASS
*Mar  1 03:14:37.875: Se2 PAP: I AUTH-ACK id 7 Len 5              <- Server verified the username/password with AUTH-ACK
                                                                        One-way authentication is complete at this point.

----->
Varies outputs from DIFFERENT debug session using : "debug ppp authentication" when using CHAP.

                                                                   >>> I = Incoming, and O = Outgoing <<<
*Mar  1 Se1 PPP: Phase is AUTHENTICATING, by both                 <- Indicates two-way authentication
*Mar  1 Se1 PPP: Phase is AUTHENTICATING, by the peer             <- Indicates a routers is performing one-way authentication
*Mar  1 Se1 PPP: Phase is AUTHENTICATING, by this end             <- Indicates a routers is performing one-way authentication

*Mar  1 Se1 LCP: I TERMREQ                                        <- Both indicates a peer is failing to authenticate the local
*Mar  1 Se1 CHAP: I FAILURE                                          router's username and password. Usually misconfiguration

*Mar  1 Se1 CHAP: O FAILURE id 10 len 26 msg is "Authentication failure"
*Mar  1 Se1 LCP: O TERMREQ [Open] id 22 len 4                     <-  The local router has failed to authenticate a peer.
                                                                        Username is misconfigured.
*Mar  1 Se1 CHAP: Username maui-soho-01 not found
*Mar  1 Se1 CHAP: Unable to validate Response.  Username R2 not found
*Mar  1 Se1 CHAP: Unable to authenticate for peer                 <- The username supplied by the peer is not configured locally

*Mar  1 Se1 CHAP: O FAILURE id 16 len 25 msg is "MD/DES compare failed" <- This error indicates a password mismatch.
```



-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

```
- OER/PfR
      + Master Controller
      + Border Routers
      + Interface Types
      + 5 Phases
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================*
  Routing Decisions
*====================*
- Refer to the following link for a flow-chart:       http://blog.ru.co.za/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 between OSPF routes, then O > O*IA > E1 > E2.
 > If a tie between BGP routes, then Bestpath Selection process.

- The default AD values can be changed with the "distance" command, but note that is it different between protocols:
 > Generic        - distance {distance}
 > EIGRP          - distance eigrp internal-distance external-distance
 > OSPF           - distance ospf {external} {inter-area} {intra-area}
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  Switching Paths
*=====================*
- 3 Steps:
 > 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.
 > Is enabled by disabling CEF/Fast switching at the interface level.
 > To enable use the following command: 'no ip route-cache' and disable CEF with '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 time consuming when huge numbers of potential
       destinations are involved.
 > To avoid this, a pre-build cache was needed, and thus CEF was born.

- With CEF (Cisco Express Forwarding), there are two main data structures:
 > The Adjacency-table:
      >> Is 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 invers-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 it 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 you ping a IP address local to the router:
 > The packet does NOT cross the backplane between interfaces. The router actually sends the packet out the interface
       to the other end.
 > So, if the interface to a peer is down, the ping will be unsuccessful.
```

```
----------
 COMMANDS
----------
# sh adjacency [detail]                              - Shows the layer2 adjacency table
                                                     - [detail] Option 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                                 - Hidden CEF command that shows load-balancing hash algorithm table

# debug arp                                          - Shows the arp query and responses
# debug ip routing                                   - Shows the routes install/removed from the routing table along with protocols
# 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 there is no route to the source or to the destination
                                                     - 'encapsulation failed': layer2 resolution is not available


#ip cef                                              - Enables CEF globally
#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
*=====================*
#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 recursive to an exit interface. (in the routing table)
- If the next-hop IP is on a multipoint interface, layer3 to layer2 resolution is required.
- If the next-hop IP is on a point-to-point interface, layer3 to layer2 resolution is NOT required.
- A exit-interface is generally used on broadcast medium 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.
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   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 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 a 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(config)# sh cdp neighbors                        - Confirm the expected stubs are CDP neighbors
#hub(config)#router odr                               - Enables ODR on the HUB router


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Secondary IP addresses
*==========================*
- The big nasty that shows poor address space planning.
- One thing to keep in mind is that all traffic generated by the router out a interface, will have the primary IP as a
      source, not the secondary.

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


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Floating Static
*=====================*
#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 recursive lookup provides end-to-end next-hop reachability, information above would work.
- If there are no end-to-end next-hop reachability, it could create a black hole.
- Assume the first route is going across a frame-relay multipoint interface with multiple DLCI's configured,
 if the DLCI for 10.1.4.0 fails on the primary remote end, but the other local DLCI's stays 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, ie successful, disconnects, timeouts, busy, errors
# sh track 5                                          - Shows the tracker based on the RTR

#ip route 10.1.4.0 255.255.255.0 10.1.3.1 90 track 5  - Primary route: If object 1 goes down, this route will be removed.
#ip route 10.1.4.0 255.255.255.0 10.1.2.1 95          - Backup route

#track 5 rtr 1                                        - Creates a track that calls RTR (Response Time Reporter) / IP sla monitor


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




*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Backup Interface
*=====================*
- Tracks the local line protocol of "primary" interface.
 > If the line is up, the "backup" interface is in standby.
 > 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.
 > Fallover, specifies the delay before the standby link gets brought up after the primary link failed.
 > Failback, specifies the delay after the primary link came 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                                           - Displays the interfaces and the status
# debug backup                                        - Displays 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
*=====================*
- Generic Route Encapsulation is a layer3 VPN technology.
- Uses IP transport protocol 47.
- Used to transport payload protocols over IPv4 network.
- GRE is payload independent. Supports both IPv4 and IPv6.
- Tunnel destination must never recurse to the tunnel interface itself.
 > This 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
        > IP
         > Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4T
          > Part 7: Protocol-Independent 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)

- MATCH options:
      as-path                  Match BGP AS path list.
      community                Match BGP community list.
      extcommunity             Match BGP/VPN extended community list.
      interface                Match first hop interface of route.
      ip                       IP specific information like ACL, prefix-lists, next-hop and route-source.
      length                   Packet length.
      local-preference         Match BGP local preference for route.
      metric                   Match metric of route.
      route-type               Match external-(BGP,EIGRP, OSPF), internal-(OSPF), local, and nssa-external route-types.
      source-protocol          Match source-protocol of route.
      tag                      Match tag of route.
```

```
- SET options:
      as-path                 Prepend string for a BGP AS-path attribute.
      community               BGP community attribute.
      dampening               Set BGP route flap dampening parameters.
      extcommunity            BGP extended community attribute.
      interface               Using a output interface for packets. Can only be used on Point-to-point links, NOT multipoint links.
      default interface       Specifies the output interface for the packet, if there is no explicit route for the destination
      ip next-hop             Specifies the next hop to which to route the packet.  (Note: It must be an adjacent router)
      ip default next-hop     Specifies the output interface for the packet, if there is no explicit route for the destination.
                                        (Note: It must be an adjacent router)
      ip next-hop verify-availability     Causes the router to confirm that the next hops of the route map are CDP neighbors
                                        of the router. Could be used IP SLA.
      local-preference        BGP local preference path attribute.
      metric                  Metric value for destination routing protocol.
      metric-type             Type of metric for destination routing protocol.
      origin                  BGP origin code.
      tag                     Tag value for destination routing protocol.
      weight                  BGP weight for routing table.


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 150.1.6.6 255.255.255.50
|     !
|     ip local policy route-map LOCAL_POLICY          - Apply the policy for router generated traffic
|


----------
 COMMANDS
----------
#show 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 was designed for point-to-point links.
- This leaves one bit 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) is still used with 31-bit prefixes.
- But directed broadcasts 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 a 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 an 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
+------------------------------------------------------
|     int Ethernet0
|      ip add 172.16.10.254 255.255.255.0
|     !
|     interface Serial 0
|      ip unnumbered Ethernet 0                             - Configures Serial0 to "borrow" Ethernet0 IP address
|     !
|     !
>     #show ip interface brief
>      Interface    IP-Address      OK?    Method   Status   Protocol
>      Ethernet0    172.16.10.254   YES    manual   up       up
>      Serial0      172.16.10.254   YES    manual   up       up



-----------
 COMMANDS
-----------
#ip unnumbered {interface}                            - Configures a 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 access-list 3          - IF the ACL matches
|      set metric 50                - THEN set
|     !
|     route-map NAME 20             - Else look at instance 20
|      set metric 20
|
```

- With redistribution, when a route is matched against a route-map instance:
 > And the instance has a "permit" parameter, the route will be redistributed.
 > And the instance has a "deny" parameter, the route is not redistributed.
 > 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 {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}
 > Defines the destination database             #set level {stub|backbone}
 > Sets the route's metric                      #set metric {value}
 > Sets the unitless tag value                  #set tag tag-value
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==================================*
   Redistribution Overview
*==================================*
- 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 which are in the router’s current IP routing table (RIB).
 > Before enabling any redistribution, make sure each protocol have full reachability within itself.
 > The metric assigned using the "redistribute metric" command takes precedence over metrics assigned with the
      "default-metric" command.

- When 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.

- When 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 ONLY looks 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 what 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(eg OSPF) to a higher AD protocol(eg 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 (eg OSPF)
           to have a AD higher, say 130 than the HIGH AD protocol (eg RIP).
      > Filter the redistributed high AD routes from being fedback via the low AD protocol, into the origination 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(eg 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 from going back into RIP.

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

- "no redistribute {protocol} metric 10 route-map NAME"
 > This command will NOT remove the actual command but only the options, ie metric & route-map.
 > To remove this redistribute command completely, specify above line without the options, eg "no redistribute {protocol}"


*---------------------*
      RIP
*---------------------*
- When redistributing any protocol into RIP, the metric must be specified.
- This can be done by
 > Specifying the 'metric' keyword.
 > Using the "default-metric" command for RIP.
 > Using a route-map.


*---------------------*
      EIGRP
*---------------------*
- With EIGRP<->OSPF mutual redistribution, by default you won't have any feedback, because
 > OSPF routes have an AD of 110
 > EIGRP routes have an AD of 90
 > EIGRP redistributed routes have an AD of 170.
- The problem comes in when there are external EIGRP routes (D EX) in the EIGRP domain with an AD of 170.
- When they get redistributed into OSPF, and back into EIGRP, traffic to those external routes from the EIGRP domain originally,
      could prefer a path via OSPF with an AD of 110, causing a loop.
- So, before redistributing EIGRP into any protocol, check the routing table for D EX routes.

- This 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 could often assist to see where the
      route was redistributed, if you use 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, it can only be changed
      for ALL external EIGRP routes(D EX).
- EIGRP requires the "no auto-summary" command else classful subnets will be redistributed.
```

```
*---------------------*
      OSPF
*---------------------*
- By default, external routes are redistributed into OSPF as external type 2 (E2) routes, with a metric of 20.
- The 'subnets' keyword is a requirement with OSPF else 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 out an AS exist.

- 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 if there are 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 candidate to be redistributed. This is a blackhole safeguard.
 > This can be disabled by using the command "bgp redistribute-internal"

- BGP routes originated through the 'network' command has a origin code of 'i-igp'
- BGP routes originated through redistribution has a 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 a 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 {ASN}                                     >>> 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

- Good link to read with great examples : http://tinyurl.com/ie-Pfr

- 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 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, but OER/PfR does utilizes 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, for example one router with two exit interfaces.
- As long as there are 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 forwarding path, but must be reachable by BRs.


- BRs (Border Routers)
 > Are the edge routers with one or more exit links to an ISP or another WAN.
 > Reports prefix and exit link measurements to the MC.
 > Enforces policy changes from the MC, by injecting a preferred route 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
 > Are the interfaces between the MC and the BRs.
 > Are used for OER communication and for passive monitoring.
 > At least one internal interface connecting to the inside network per border router is required.

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

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

- OER communication between the MC and the BRs are carried separately from routing protocol traffic.
- From IOS 12.4(9)T, the ability to monitor and control inbound traffic was introduced.

- Prefixes or traffic classed pass through different states after they are learned.
- The traffic-class states are as follows:
 > 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 holddown state. This is intended to prevent flapping.
 > In-Policy: The status of the prefix matches policies. No changes are made when 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 found the MC will use the best one.

- 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 are called a 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 profiles interesting traffic, which has to be optimized by learning the flows as they pass through the router.
 > Non-interesting traffic will be ignored.
 > BRs sorts traffic based on delay and throughput and sends 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, Layer 4 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          - are based on TCP RTT (Round Trip Time) (initial SYN to the following ACK).
                  => Packet loss    - by tracking TCP sequence numbers for each TCP flow.
                  => Reachability   - by tracking TCP-SYNs that wasn'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          - are based on TCP RTT (initial SYN to following ACK).
                  => Reachability   - by tracking TCP-SYNs that didn't receive TCP-ACKs.
                  => Jitter         - measuring the variable delay between packets arriving at the destination.
                  => MOS            - a 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, where 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 seconds.



*----------------------------------*
    Phase 3 - Apply Policy Phase
*----------------------------------*
 > By default, OER runs in an observe mode during the profile, measure, and apply policy phases (no changes to network are made
      until 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     - are defined for prefixes or for applications.
      >> Link policies              - are 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                    - is the network traffic sent to the specific traffic class entry.
      >> An action                  - is a routing table change.
      >> A triggered event          - is 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 that 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
            -> Is the minimum period of time that a new exit must be used before an alternate exit can be selected.
            -> Used to prevent the traffic class entry from flapping because of rapid state changes.

      >> Periodic Timer
            -> Is how the MC will try 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 best delay while the other has 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.
 > Variance configures the acceptable range (%) 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 initiates route changes when one of the following occurs:
      >> A traffic-class goes OOP,
      >> An exit link goes OOP,
      >> Periodic timer expires and the select exit mode is configured as select best mode.
 > A measured prefixes' 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 depends 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. (IGP's)

      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 does 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.
```

```
      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 (out-of-profile) the previous optimizing changes will be reverted.


- 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 fines the key-chain to be used later
|     !
|     oer master
|      logging                                  - Enables syslogging
|      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 perdiod 8                       - The amount of time the router will learn prefixes
|       periodic interval 15                    - The time between the learning periods
|       !
|      border 172.17.100.1 key-chain KEY1       - Define 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 172.104.1 key-chain KEY1
|       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
|
```

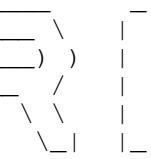
```
>>BR1 CONFIGURATION
|
|     key-chain KEY1
|      key 1
|       key-string pfr
|     !
|     oer border
|      master 172.17.10.1 key-chain KEY1
|      local fa0/0                              - Define the Local interface used to communication
|      active-probe address source int fa0/0    - This BR will source active probes from Fa0/0
|     !
|
>>BR2 CONFIGURATION
|
|     key-chain KEY1
|      key 1
|       key-string pfr
|     !
|     oer border
|      master 172.17.10.1 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                              - Displays policy information, ie timers, next-hop etc
# sh oer master prefix                              - Displays the status of the monitored prefixes
# sh oer master link-group                          - Displays information about configured OER link groups
# sh oer master traffic-class                       - Displays information about traffic classes that are monitored and controlled
# sh oer border                                     - Displays detailed info about the BR and connecting MC
# sh oer border passive learn                       - Displays traffic class filter and aggregation ACL information
# sh oer border passive cache                       - Displays real-time prefix information collected from the BR
# sh oer border passive prefixes                    - Displays the passive measurement information collected by netflow
# sh oer border active-probes                       - Displays connection status, info about active probes
# sh oer border routes {bgp | static}               - Displays 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 prefix OR use acl-filter
  #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 an 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} {sequance}                            - 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} {sequance}                    - 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 {valule}            - 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 an 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
```



-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

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

- If a router must send an update with more than 25 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 a secondary IP addresses on a interfaces as separate data links, and can exchange routes with  a secondary IP. But take
      note thatall traffic generated by a router will always have the primary address as a source.
- RIP performs a soure-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.
- Another instance where it might be needed to disable the source-validation, is when the source address is on a different subnet
      that, the locally configured address, ie local has a /32 and remote side a /24. This can be seen with "debug ip rip events".

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

-----------
 COMMANDS
-----------
# ping 224.0.0.9                                       - All RIPv2 enabled routers will answer the ping and respond
# debug ip rip events                                  - Displays 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 & Timers
*====================================*
- RIP uses hop-count as a metric.
- 1 hop per interface.
- 16 hops = unreachable

- Update timer (30sec)  - A router sends a response message out every RIP-enabled interfaces every 30 seconds on average.
- Invalid timer (180sec)- Amount of time a route can stay in the routing table without being updated.
- Holddown (180sec)     - An update with a hop count higher that the metric recorded in the table will place a route in holddown.
- Flush timer (240sec)  - The time when a invalid route get 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.
- 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


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
   Updates Types
*====================================*
- Broadcast
 > Default for RIPv1
 > RIPv2 optional
      #ip rip v2-broadcast                          - Enable broadcasts at a interface level for RIPv2

- Multicast
 > RIPv2 default to 224.0.0.9
      #version 2                                    - Enables RIP version 2

- Unicast
 > RIPv1/RIPv2 optional
      #neighbor {IP}                                - Under the process, send unicast updates to neighbor
                                                    - Useful on NBMA networks like frame-relay
                                                    - This does not stop the sending of broad/multicasts, use
                                                          "passive-interfaces" for that


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

#router rip
 #network {ip}                                      - Assign the classful network to match interfaces to be advertise by RIP.
```

-=-

Passive interfaces

- "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 advertises 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.

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

```
#ip rip triggered                              - Enabled triggered updates
                                               - Only available on serial link, 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 the summary 10/7 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
                                                      - It's subject to a subnet of this aggregate being 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 required to match the subnet mask, 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: Offset-list Example
+-------------------------------------------------------
|     access-list 1 permit 10.33.0.0 0.0.0.0              - Identifies the route entry for subnet 10.33.0.0
|     !
|     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 10.0.0.0/8
|     ip prefix-list SOURCE permit 1.2.3.4/32
|     distribute-list prefix ROUTE gateway SOURCE in        - Only accept 10.0.0.0/8 route from 1.2.3.4
|


CONFIG-SET: Extended Access-list Example (same as using prefix-list)
+-------------------------------------------------------
|     access-list 100 permit ip host 1.2.3.4 host 10.0.0.0
|     distribute-list 100 in                               - Only accept 10.0.0.0/8 route from 1.2.3.4
|


-----------
 COMMANDS
-----------
#offset-list [ACL] {in|out} {offset} {Interface}      - Adds offset from RIP metrics

#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
*====================================*

# default-information originate                       - Generates and advertises an unconditional default route to neighbors

# ip prefix-list DEFAULT seq 5 permit 0.0.0.0/0
# distribute-list prefix DEFAULT out                  - Limit the advertisements to only the default route sent out
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
   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 interfaces 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 ethernet 0
  #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                   >>>   {} curl-brackets indicates replaceble values      <<<
*====================================*
*******************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*******************************************************************************************************************************

- When troubleshooting RIP updates and route-selection issues, consider the following:
 > Are the necessary RIP interfaces in a '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 neighbors 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 usage 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
 > Does routes redistributed into RIP have a valid hopcount defined?                       # sh run | i redist|default.*met
 > If authentication is configured, does 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

- 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/IP's.

- Route entries are classified into one of three categories:
  > Interior routes: Is 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 same AS as interior route because it falls within the same major network.

```
 > System routes: Is a path to a network address, which 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: Is 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 a exterior route.



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
  Metric, Timers and K-values
*==============================*
- EIGRP calculates all metrics from outgoing interfaces only.
- The composite metric is minimum bandwidth of an outgoing interface, cumulative delay, load, and reliability.
      and smallest MTU along a path.
- Metrics:
 > BW        - Expressed in units of kilobits per second.
             - Static number used for metric calculation only, doesn't reflect actual bandwidth.
             - To calculate EIGRP bandwidth metric amount: 10 000 000/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 eight-bit number, where 255=100% reliable link and 1= minimally reliable link.

 > LOAD      - Dynamically measured.
             - Is expressed as an eight-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 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 e0
 #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, to load balance.
- Variance must 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.

- For a good load-sharing example refer to : http://blog.ru.co.za/2009/04/02/eigrp-metric-manipulation/
- Load-sharing by default is based inversely to the traffic-share rate among the multiple paths.
 > So if wanting a traffic-share rate of 1:5, the first path would get five times more traffic than the second.
- This can be changed to use only the best metric path, even though both routes are in the table with
      #traffic-share min across-interfaces

-----------
 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 15
 #variance 5                                    - Meaning a interface metric can be up to 5 times more than the current FD (def=1)
 #maximum-paths 2                               - Changes the default (4 paths) over which EIGRP can be set load balance.
 #traffic-share balanced                        - Default: share inversely proportional to metric
 #traffic-share min across-interfaces           - Only use the best metric path, even though multiple in routing table


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===========================*
  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 with each timer

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

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

- Can be sent as unicast at a process level with the "neighbor" command.
- But both sides must be configured to use unicast.
- BEWARE: If configured, EIGRP stops processing all multicast packets that come inbound on that interface.
       Also stops sending multicast packets on that interface.
- BEWARE: Upon configuring all sessions from that interface will be dropped.

- Using an ACL to filter EIGRP traffic between two neighbors are 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              - Permit all other traffic
#int 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 use 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 was not acknowledged by the neighbor it was 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 feasible distance (FD) of that destination.
- The feasibility condition (FC) is a condition that is met if a neighbor's advertised distance (AD) 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 feasible successor (FS) for that destination.
- Because feasible successors are always "downstream," a router will never choose a path that will lead back through itself, thus
     creating a loop.
- Such a path would have a distance larger than the FD.
- Every destination for which one or more feasible successors exist, will be recorded in a topology table.
- Each route after inserted, when no diffusing is taking place will be in a passive state.
- If there are two successors with the 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 a feasible successor.
- If a 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 a feasible successor 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 stuck-in-active (SIA).
- At the completion of the diffusing computation, the originating router will set FD to infinity to ensure that any neighbor
      replying with a finite distance to the destination will meet the FC and become a feasible successor.
- 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)

# show 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.

#passive-interface {int}                        - Disables the interface from sending hellos
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===========================*
  Split-Horizon
*===========================*
- Is always enabled with EIGRP.
- Remember to disable split-horizon with physical multi-point frame-relay interfaces.

-----------
 COMMANDS
-----------
#interface eth0
 #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 keychain, 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 Serial0
 #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 classful boundary when passing major network boundary, but this can be disabled.
      If auto-summary is enabled, interfaces are summarized at class boundary.
- A route to Null0 for summary routes are created to prevent black-holes.
- Disabling automatic summarization can prevent ambiguous routing between similar network subnets. (It is always recommended)

- Manual summarization for EIGRP is interface-specific.
 > This provides the flexibility to be able to advertise different summary routes out 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)
#int E0
 #ip summary-address eigrp {ASN} {aggregate} [leak-map] [AD]
                                                - Will automatically suppress 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 + 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 metric.

#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 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

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

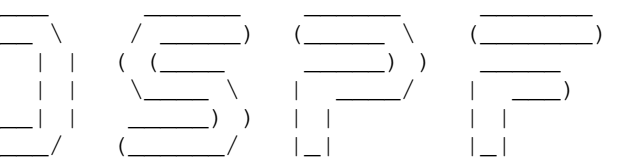
```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Stub Routing
*==========================*
- A router that has EIGRP stub neighbors will not send queries to those stubs, thereby eliminating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

-----------
 COMMANDS
-----------
# show ip eigrp neighbors [detail]              - With detail option: (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
                                                - DEFAULT: Only updates containing connected and summary routes will be sent
                                                - Receive-Only]: The stub router will not send any route information in updates


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Bandwidth Percent
*==========================*
- EIGRP is designed to use no more than 50 percent 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 percent 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 Serial0                              - Assumes the physical clock is 512k
 #bandwidth 128
 #ip bandwidth-percent eigrp 1 200              - Adjusts the EIGRP bandwidth percent to 200% of 128K
                                                - Which is 256K, half of the actual link bandwidth 512k
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Troubleshooting EIGRP            >>>   {} curl-brackets indicates replaceble values      <<<
*==========================*
*****************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*****************************************************************************************************************************

- When examining an individual router's configuration, consider the following:
 > Are the necessary EIGRP interfaces in a '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 anothers 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 IP's.       # 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? (It is required to match)                 # 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 usage of "passive-interface" preventing a neighbor adjacency?                   # sh run | i passive
 > If authentication is configured, does 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 enqueued 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.                        # 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 in 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 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 a 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 a stub? (look for CONNECTED SUMMARY)   # sh ip eigrp neighbors detail
 > Does 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
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        86       2/95          95           0
Se1/1              1        0/0        34      10/380        580           0
Lo0                0        0/0         0       0/10           0           0

----->
#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
Se1/1              0        0/0         0      10/380        580           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 17/19
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 5
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP"
```

```
----->
#sh ip eigrp traffic
IP-EIGRP Traffic Statistics for AS 100
  Hellos sent/received: 835/816
  Updates sent/received: 36/33
  Queries sent/received: 3/5
  Replies sent/received: 5/3
  Acks sent/received: 23/29
  Input queue high water mark 3, 0 drops
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 210
  PDM Process ID: 209


----->
#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 dst
        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

- 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 {no} show the Retransmission retry number, amount of re-attempts to send updates, without acknowledgements
 > RTO show 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

- This command is not IP specific, but instead shows EIGRP neighbor events
- Optionally, the AS-Number and the neighbor IP could be specified to filter the output.
```



-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

- Authentication
  + Area
  + Interface
  + MD5
  + Clear Text
  + Null
  + Virtual-Link
- Default Routing
  + Always
  + Conditional
- OSPF Demand Circuit
- Troubleshooting OSPF
- Output-101

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

OSPF Overview

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

- Uses own transport protocol: 89
- Supports equal-cost load balancing for more efficient use. (The amount is not an OSPF limitation instead this set by the vendor depending on their hardware platform. For most IOS versions this is limited to either 6 or 8 paths).
- 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 and therefore will not send hellos on them.
  > Consequently, no adjacencies can be established on secondary networks.

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

Hello Protocol

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

- The hello protocol serves several purposes:
  > It is the 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 when a neighbor sees his own router ID in a received hello.
  > It elects Designated Routers (DRs) and Backup Designated Routers (BDRs) on Broadcast and NBMA networks.

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

- Router Dead-Interval
  > Is the period of time to elapse, if a router does not receive a hello from a neighbor, before declaring that neighbor down.
  > Cisco default is four times the hello-interval but can be changed with the command "ip ospf dead-interval" below.

```
- 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.

- 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 second.
 > This is achieved by using the "ip ospf dead-interval minimal" command. Setting the dead interval to 1 second.
 > The hello-multiplier value is set to the number of hello packets you want sent during that 1 second.
 > Example: #ip ospf dead-interval min hello-multiplier 5   - hellos are sent 5 times per/sec, thus at a 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 interface
# sh ip ospf interface                            - Displays OSPF-related interface information, DR, BDR, etc
# sh ip ospf interface brief                      - Shows brief summary of which interface is running which ospf areas

#interface ser0
 #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


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================*
   Network Types
*====================*
- An OSPF router maintains a data structure for each OSPF-enabled interface.
- If you change the network type, you will also change the hello and dead timers accordingly.
```

- OSPF defines six network types:
 > Broadcast networks
      >> Default network on ethernet and FDDI.
      >> 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 networks
      >> Can connect more than two routers but have no native broadcast capability.
      >> Non-Broadcast is the default network type on multipoint frame-relay interface, eg 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 networks
      >> Default on T1, DS-3, or SONET links, point-to-point sub-interface on frame-relay and ATM networks.
      >> Uses the multicast destination to AllSPFRouters (224.0.0.5), except for retransmitted LSAs which are unicast.
      >> NO DR/BDR election, OSPF configured as per normal.
      >> 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 networks
      >> Cisco proprietary, and not a default option, but best choice for NBMA networks.
      >> Are special configurations 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 instead
            of the actual networks. ie /32 in the routing table.
      >> 30 hello / 120 dead-interval.

 > Point-to-multipoint non-broadcast networks
      >> Cisco proprietary, 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.
             This command is only required on the one side.
      >> 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.
      >> Was created to allow for the assignment of the cost per neighbor appose to 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
      >> Are 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 cannot 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 neighbouring 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.
      >> To see the cost of using the transit area use "sh ip ospf virtual-link" and refer to 'cost of using'.
      >> The cost of the virtual link is the cost of the route to the neighbors interface via the transit area.

 > OSPF over GRE
      >> OSPF virtual links cannot 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 a IP address with a network statement in area0.

 > Stub/loopback networks
      >> 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 interface                          - Displays OSPF-related interface information, DR, BDR, 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                     - Change the network type to broadcast. Timers: 10/40
 #ip ospf network non-broadcast                 - Change the network type to NBMA. Timers: 30/120. Require manual neighbors
 #ip ospf network point-to-point                - Change the network type to point-to-point. Timers: 10/40
 #ip ospf network point-to-multipoint           - Change the network type to point-to-multipoint. Timers: 30/120
 #ip ospf network point-to-multi [non-broadcast]- Change 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,
   but the cost from the pseudonode to any attached router is 0.
- 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 on Cisco routers are 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 a ‘UP’ state. If no loopbacks are configured, it is the highest interface IP in a ‘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 a 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 of Data Descriptor Packet's (DDP's).
>> The neighbor with the highest router ID becomes the master.

> Exchange
>> The router sends DDP's 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 DD packet are used to manage the adjacency building process:
*> I-bit, or Initial bit, when set indicates the first DD packet sent.
*> M-bit, or More bit, when set indicates that this is not the last DD packet to be sent.
*> MS-bit, or Master/Slave bit, is set in the DD packets 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 - A 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 - Are routers whose interfaces all belong to the same area. These routers have a single link-state database.
 > Backbone - Are 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.
            - Has at least one interface, which belongs to the backbone, and must maintain a separate link-state database for
                each of its connected areas.
 > ASBR     - Is a gateway to external traffic. It injects routes into the OSPF domain that was learned (redistributed)
                from another external protocol.



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==================================*
   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.
      > Lists all of a router's links, or interfaces, the state and outgoing cost of each link, and any known OSPF
            neighbors on the link.
      > Local area flooding scope.
      > Describes 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.
      > Lists all attached routers, including the DR itself.
      > Local area flooding scope.
      > Describes who is 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.
      > Inter-Area flooding scope.
      > Describes 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.
      > Inter-Area flooding scope.
      > Describes who 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.
      > Autonomous system wide flooding scope.
      > Describes 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 generates syslog messages if such packets are received.
      > It might be necessary to configure a router to ignore these packets and to prevent a large number of syslog messages
      > Configured with "ospf ignore lsa mospf"

 7 NSSA External LSAs
      > Are originated by ASBRs within not-so-stubby areas (NSSAs).
      > Similar to an AS External LSA, except NSSA External LSAs are flooded only within the not-so-stubby areas
            in which it 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
      > Have been used to add various extensions to OSPF, such as traffic engineering parameters for MPLS networks.


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


- OSPF LSA Throttling
 > Provides a dynamic mechanism to slow down link-state advertisement (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 database database-summary          - Displays 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}
 #timers pacing lsa-group                        - Allows more LSA's to be grouped together before being flooded (default=4min)
 #timers lsa-group-pacing {seconds}              - Changes the group pacing interval of LSA's
 #timers spf {spf-delay} {spf-holdtime}          - Changes the delay time between receiving a topology change and SPF calculation
 #ospf ignore lsa mospf                          - Ignore MOSPF LSA packets, stops generating syslog messages.
 #neighbor {ip} database-filter all out          - Block the flooding of OSPF LSA packets only to a specific neighor

 #max-lsa {max-no} [threshold-%] [warning-only] [ignore-time] [ignore-count] [reset-time]
                                                 - {max number}: of non-self-generated LSA’s that can be kept in the OSPF LSDB
                                                 - [threshold]: Percentage at which a warning message is logged. The default is 75%
                                                 - [warning-only]: OSPF process never enters ignore state. (Def = Disabled)
                                                 - [ignore-time]: time to ignore neighbors after the limit's exceeded. (Def = 5 min)
                                                 - [ignore-count]: number of times consecutively to enter ignore state. (Def = 5)
                                                 - [reset-time]: time before ignore count gets reset (Def = 10 min)

 #timers throttle lsa all {start-interval} {hold-interval} {max-interval}
                                                 - Sets the rate-limiting values (in milliseconds) for LSA generation
                                                 - {start-interval}: (Def = 0 ms)
                                                 - {hold-interval}: (Def = 5000 ms)
                                                 - {max-interval}: (Def = 5000 ms)

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

-=-

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

Area Types

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

- LSA filtering is done in two ways:
  > Area Types
  > LSA 3 filtering see filtering below

- When only a single area network is used, it doen 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 is more than one area.

> Stub Areas
  > Type 4 ASBR summary LSAs and 5 AS External LSAs are not flooded into stub areas.
  > Still receives inter-area type 3 LSA's.
  > ABRs at the edge of a stub area use network summary (type 3) LSAs to advertise a single default route (0/0) into the area, for destination 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" command.
  > Configured on ALL routers in the stub area with "area stub" command.
  > Stub area restrictions:-
    >> All routers in a stub area must have identical link-state databases, and agree to be stub.
    >> To ensure this condition, all stub routers will set a flag (the E-bit) in their hello packets to zero. 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
    > Uses a default route to reach not only destinations external to the autonomous system but also all destinations outside the area.
    > The ABR of Totally Stubby Area will block all summary LSAs with the exception of a single type 3 LSA to advertise 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)
    > An area that allows redistributed while retaining the characteristics of a stub area to the rest of the AS.
    > No type 4 and 5 LSA's, but redistribution is allowed, ie AS-external routes.
    > The ASBR in an NSSA will originate type 7 LSAs to advertise the external destinations.
    > These NSSA external LSAs are flooded throughout the NSSA but are blocked at the 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 one, the type 7 LSA translates into a type 5 LSA before being flooded to other areas.
    > If the P-bit is set to zero, no translation will take place and the destination in the type 7 LSA will not be advertised outside of the NSSA.
    > Configured on ALL routers in the NSSA area with "area nssa".
    > Biggest difference to a stub area, redistribution is allowed, and no default route by default is sent into the area.
    > With NSSA, the ABR does not automatically originate a default route.
    > To originate a default into a NSSA area, use the command "area nssa default-originate".

> Totally NSSA
    > A ABR makes an NSSA totally stubby, with 'area nssa no-summary'.
    > Removes inter-area (type 3) LSA's.
    > Removes external (type 4 and type 5) LSA's.
    > The ABR originates a default as 'O*IA'.
    > Configured with 'area nssa no-summary', which is necessary only at the ABR; the internal routers use the standard NSSA area configuration.
    > Allows redistribution into NSSA (LSA 7).

- 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 the 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 only explicitly filters type 5 LSAs, but it implicitly filters type 4 LSAs as well as there is no need for the ABR to generate a type 4 LSA.

- When an ABR is also an ASBR and is connected to a NSSA, the default behaviour 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, but use the 0.0.0.0 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 forwarded traffic to the translating NSSA ASBRs.

```
-----------
 COMMANDS
-----------
 #router ospf 99
  #area 1 default-cost {cost}                    - Changes the cost of the default route advertised by the ABR. (default = 1)

  #area 1 stub                                   - Configures attached area 1 as a stub area, required 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 as a totally stubby area, only needed on ABR's
                                                 - 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 nssa, required on all area routers
                                                 - NO default route is automatically generated.

  #area 4 nssa default-information-originate     - Configures attached area 4 to nssa, only needed on ABR's 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, only needed on ABR's
                                                 - 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
                                                 - Will show 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
                                                 - Will show 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 LSA's
                                                 >> [no-summary] blocks type 3 LSA's except the default route type 3 LSA
                                                 >> [nssa] blocks type 4 and type 5 LSA's, but allows type 7 redistribution
                                                 >> [no-redistribution] blocks type 7 LSA's
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Filtering
*=====================*
- Filtering can only occur between areas, by RFC standard: 'All routers within a area must have the same link-state database'.
- Different ways to filter traffic:
 > With a "filter-list".
 > With a "distribute-list" referencing a ACL|prefix-list|route-map.
 > With the "distance" command.
 > With the "area range" command (see summary section below).
 > With the "summary-address" command on a NSSA ABR for external prefix filtering (see summary section below).


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


- Distribute-list
 > Note that distribute-lists ONLY blocks routes from entering the LOCAL RIB, it DOES 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       external route (BGP, EIGRP and OSPF type 1/2)
      >>  internal       internal route (including OSPF intra/inter area)
      >>  local          locally generated route
      >>  nssa-external  nssa-external route (OSPF type 1/2)


-----------
 COMMANDS
-----------
#ip prefix LIST1 seq 10 deny 192.168.1.0/24     - Matches 192.168.1.0/24 exactly to be denied
#ip prefix LIST1 seq 20 permit 0.0.0.0/0 le 32  - Permits everything else

#router ospf 1
 #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 , ie don't send 192.168.1.0
                                                - 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 192.168.1.0 from entering the RIB, but it's still in LSA-DB
 #distance 255 192.168.1.5 0.0.0.0 99           - Assign admin distance 255 for routes matching ACL-99 from src 192.168.1.5
 #distance ospf {external | inter-area | intra-area}  - Change the distance of OSPF routes
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   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 supported by OSPF:
 > Inter-area summarization
      >> Used for summarization of internal OSPF area routes at ABRs.
      >> A route to Null0 will be entered automatically, but this can be disable with "no discard-route".
      >> The "area range" command specifies the area to which the summary address belongs.
      >> The default behaviour 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 prefixes as they move between areas.
      >> Summarizes type 3 LSA's.

 > External route summarization
      >> Allows a set of external addresses to be redistributed into an OSPF domain as a summary address at ASBR's.
      >> Is configured with "summary-address" command on the ASBR's.
      >> Any more specific subnet addresses which fall within the range of the specified summary address will be suppressed.
      >> Summarizes type 5 and 7 LSA's.


-----------
 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] Do NOT advertise more specifics
                                                - [cost] User specified metric for this range

 #summary-address 160.1.60.0 255.255.255.0 not-advertise
                                                - Summarizes type 5 and type 7 LSA's
                                                - Any more-specifics which are within the range will be suppressed
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
  Stub Router Advertisement
*==========================*
- Do not confuse this with STUB AREAS.
- Are updates 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.
- Optionally the feature can be used to allow the router to advertise a maximum metric until the BGP routing table converges.
- Typical scenario when used, is when there is multiple links between two areas, and one link should only be used as a last resort.

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

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 = seconds
|                                               - There is no-default, (value = 5-86,400 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 the specified interface.
- This work differently to distance vector protocols like RIP, where routes will still be received, but not sent.
- To get the same effect of a distance vector protocols passive interface in OSPF,(ie. receive routes but don't send routes) use:
      "ip ospf database-filter all out" under the interface.

-----------
 COMMANDS
-----------
# sh ip ospf interface                              - Indicates passive-interface by "no hello"
 #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 routes
*========================*
- 3 ways to originate route with OSPF
 > "network area" command under the ospf process.
 > "ip ospf area" command under the interfaces. (Switches do not support this command)
 > Redistribution from connected interfaces, statics or other protocols.


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   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 destinations are inserted
   into the routing table.
      >> Seen by #sh ip route ospf
 > Router entries are routes to ABRs and ASBRs. This information kept is in a separate, internal route table.
      >> Seen by #sh ip ospf border-routers

- OSPF Route Table Lookups:
 1 - Longest Match
 2 - Most to Least Preferred Path Type:
      a. O        - Intra-area paths are to destinations within one of the router's attached areas
      b. O IA     - Inter-area paths are to destinations in another area but within the OSPF autonomous system
      c. E1 (N1)  - External paths are to destinations outside the OSPF autonomous system, (Ext cost + Cost to ASBR)
      d. E2 (N2)  - External paths are to destinations outside the OSPF autonomous system, (Ext cost to dst is used) !DEFAULT!
 3 - Use Lowest cost metric, unless equal-cost paths exist.
```

- OSPF external routes are, by default, E2 paths.
- E1 and N1 are cumulative metrics, the ASBR advertised cost and internal OSPF cost to the ASBR.
- E2 and N2 are static metrics as advertised by ASBR.
- Using E1 Metrics: Packets will be routed to external destinations out of the network, usually at the closest exit point.
- Using E2 Metrics: If one wants packets to exit the network at the closest point to their external destination.

- Cost is the OSPF metric, expressed as an 16-bit integer in the range of 1 to 65535.
- Cisco uses a default cost of 10^8/BW (100MB), expressed in whole numbers, where 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, Cisco provides the command "auto-cost reference-bandwidth" which allows the default reference bandwidth to be changed.
- COST = REF-BW/INT-BW

- Cost can be modified with:
 > interface "bandwidth"
 > interface "ip ospf cost"
 > process "auto-cost"
 > process "neighbor x.x.x.x cost"

- iOSPF (Incremental OSPF)
 > 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 ABR's and ASBR's

#interface s0/1
 #ip ospf cost                                  - Changes the outgoing cost for packets transmitted from the configured interface
#router ospf 1
 #ispf                                          - Enables iOSPF
 #auto-cost reference-bandwidth                 - Remedies the Cisco default reference bandwidth problem
 #neighbor {ip} [priority {pri}] [cost {cost}]  - Changes the Cost for routes received from the specified neighbor
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Authentication
*=====================*
- When authentication is configured, it should be configured for the entire area.
- The passwords do not have to be the same 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, first remove the config of the interface.

- A interface-level command will overwrite the OSPF process-level command.
- To configure type 1 authentication for an area
 > Under the interface
      #ip ospf authentication-key
 > Under the OSPF process
      #area {id} authentication

- The virtual-link command will overwrite the OSPF process-level command.
- Virtual-link authentication can be enabled in the following 2 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 1
 #area 10 authentication                         - STEP A1: Enables type 1 authentication under the process
 #area 20 authentication {message-digest}        - STEP B1: Enables type 2 MD5 authentication under the process

 #area 30 virtual-link 1.1.1.1 auth {key}        - STEP C1: Enables type 1 authentication for the virtual-link
 #area 40 virtual-link 2.2.2.2 message-digest-key {key-id} md5 {key}
                                                 - STEP D1: 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                         - STEP A2: Enables type 1 authentication
 #ip ospf authentication-key {key}               - STEP A3: Enables type 1 authentication under the interface for the area

#interface Serial2
 #ip ospf message-digest-key {key-id} md5 {key} - STEP B2: Enables type 2 MD5 auth under the interface for the area
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================*
   OSPF Demand Circuit
*=======================*
- An enhancement which suppresses the hello and LSA refresh functions, so that a link does not have to be constantly up.
- OSPF brings up a demand link up to perform the initial database synchronization and subsequently, to flood only
    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 (Hellos are used only to bring up the link),
      OSPF must make a presumption of reachability.
- Demand circuit must be a point-to-point link for OSPF.
- Command "ip ospf demand-circuit" is only needed on the one side
- Changes to the interface and neighbor state machines and to the flooding procedure:
 > MaxAge = DoNotAge
 > A new flag known as the demand circuit bit (DC-bit) is added to all LSAs it originates
 > The DoNotAge bit is set on LSAs advertised out, 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             >>>   {} curl-brackets indicates replaceble values      <<<
*==========================*
**************************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
**************************************************************************************************************************************


- 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 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
 > 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 run | i passive-interface
 > Does each router have the correct router-id? Any duplicates?                               # 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 a area type configured as the same type?                                - Normal, Stub, or NSSA
 > Are all area border routers configured correctly?                                          - Totally-stub, or totally-NSSA
 > Remember with multiple NSSA ABR's, only router with highest RID does the conversion!
 > Is there a virtual link that transits/configured within a stub area?                       - If so, configure GRE tunnel instead
 > Is there a summary LSA to leave an area for unknown subnets/AS's?                          - For NSSA manual default is needed
 > Does an external LSA exist to leave ospf domain?                                           # sh ip ospf data external
 > Is the forwarding address known as an internal OSPF route? (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
 > Are the hello/dead timers the same between neighbors?                                      # sh ip ospf int | i line|Dead
 > If different network types, are they compatible?                                           # sh ip ospf int | i line|Type
 > Are the optional capabilities 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 anothers 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 is 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 a blackhole DR? (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
 > If normal authentication, is the password the same between neighbor interfaces?       # sh run | i auth.*key
 > If md5 authentication, is the digest-key the same between neighbor interfaces?       # sh run | i digest-key
 > Do the all 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.                                         # debug ip ospf adj


- Link-state database problems. (All databases must be the same for each area)
 > Is the local router generating the expected LSA's?                                   # sh ip ospf database self-originate
 > Is the local router receiving the expected LSA's from a neighbor?                    # sh ip ospf database adv-router {ip}
 > Are any filters configured denying LSA's sent into an area?                          # sh run | i filter-list
 > Are any distribute lists configured denying entry in the local RIB?                  # sh run | i distribute-list
 > Is summarization the cause of LSA not being seen?                                    # sh run | i area range|summary-add
 > Do all the routers in a area have the same amount of LSAs?                           # sh ip ospf database database-summary
     >> If not, any interface filtering LSA sent out?                                   # sh run | i database-filter
 > Do the checksums for every LSA in the databases match between routers?               # sh ip ospf database
 > Any LSA's have a higher than others sequence number?  (Look at Seq#)                 # sh ip ospf database
     >> This could point to an unstable link, causing by frequent LSA advertising.     # sh int {int} | i error|drops
     >> Multiple LSA's with high sequence numbers could indicate a neighbor issue.     # sh ip ospf neighbor detail | i Neighbor
 > Has there been many SPF calculations? What triggered these events?                   # sh ip ospf statistics
 > Have you checked memory and CPU utilization on the routers?                          # sh process cpu history


- When doing redistribution, consider the following:
 > Is the 'subnets' keyword used in the statement?                                      # sh run | i redistribute.*subnets
```

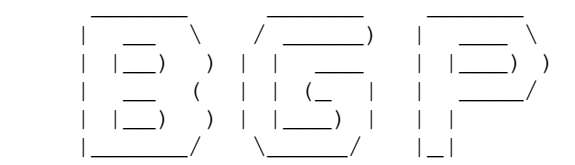
```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
      OUTPUT 101
*========================*
------>
#show ip ospf interface serial1.738
Serial1.738 is up, line protocol is up
   Internet Address 192.168.21.21/30, Area 7
   Process ID 1, Router ID 192.168.30.70, 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 192.168.30.77
   Message digest authentication enabled
      Youngest key id is 10

#show ip ospf interface ethernet1
Ethernet1 is up, line protocol is up
   Internet Address 192.168.32.4/24, Area 78
   Process ID 1, Router ID 192.168.30.70, Network Type BROADCAST, Cost: 10
   Transmit Delay is 1 sec, State DROTHER, Priority 1
   Designated Router (ID) 192.168.30.254, int address 192.168.32.2
   Backup Designated router (ID) 192.168.30.80, int address 192.168.32.1
   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 192.168.30.80 (Backup Designated Router)
      Adjacent with neighbor 192.168.30.254 (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.
- Rxmt Interval   : The period, in seconds, the router will wait between retransmissions of OSPF packets that have
                        not been acknowledged
- AuType          : Describes the type of authentication used on the network, Null, Simple Password, or Cryptographic
```

```
---->
#show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Int
192.168.30.70     1   FULL/DR         00:00:34    192.168.17.73   Ethernet0
192.168.30.254    1   FULL/DR         00:00:34    192.168.32.2    Ethernet1
192.168.30.70     1   FULL/BDR        00:00:34    192.168.32.4    Ethernet1
192.168.30.30     1   FULL/ -         00:00:33    192.168.17.50   Serial0.23
192.168.30.10     1   FULL/ -         00:00:32    192.168.17.9    Serial1
192.168.30.68     1   FULL/ -         00:00:39    192.168.21.134  Serial2.824



------>
#show ip ospf database
OSPF router with ID (192.168.30.50) (Process ID 1)

       Router Link States           <--- Type 1
Link ID            ADV Router      Age   Seq#        Checksum   Link count
192.168.30.10      192.168.30.10   1010  0x80001416  0xA818     3
192.168.30.20      192.168.30.20   677   0x800013C9  0xDE18     3

      Net Link States              <--- Type 2
Link ID            ADV Router      Age   Seq#        Checksum
192.168.17.18      192.168.30.20    677   0x800001AD  0x849A
192.168.17.34      192.168.30.60    695   0x800003E2  0x4619

      Summary Net Link States      <--- Type 3
Link ID            ADV Router      Age   Seq#        Checksum
172.16.121.0       192.168.30.40   1231  0x80000D88  0x73BF
172.16.121.0       192.168.30.50   34    0x800003F4  0xF90D

      Summary ASB Link States      <--- Type 4
Link ID            ADV Router      Age   Seq#        Checksum
192.168.30.12      192.168.30.40   1240  0x80000006  0x6980
192.168.30.12      192.168.30.50   42    0x80000008  0xC423

      AS External Link States      <--- Type 5
Link ID            ADV Router      Age   Seq#        Checksum   Tag
10.83.10.0         192.168.30.60   459   0x80000D49  0x9C0B     0
10.22.85.0         192.168.30.80   1056  0x800001F7  0x6B4B     65502


- The Router link states: LSA's are generated by each router within a area for all its connected interfaces.
- The Net Link states: LSA's are only created by the DR. 192.168.30.20 believes it is the DR for 192.168.17.18
      and 192.168.30.60 believes itself if the DR for 192.168.17.34.
- The summary Net Link states: Shows networks not local to the area
- Summary ASB Link States: Shows which router are doing the redistribution
- AS External Link States: Shows the redistributed routes.
```

```
------>
#show ip ospf database router 192.168.30.10
      OSPF Router with ID (192.168.30.50) (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: 192.168.30.10
Advertising Router: 192.168.30.10
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: 192.168.30.80
   (Link Data) router int address: 192.168.17.9
    Number of TOS metrics: 0
     TOS 0 Metrics: 64

  Link connected to: a Stub Network
   (Link ID) Network/subnet number: 192.168.17.8
   (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: 192.168.17.18
   (Link Data) router int address: 192.168.17.17
    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.
 > From the output "Routing Bit Set on this LSA," it means that the route to this destination is in the routing table.



------>
#show ip ospf database database-summary
            OSPF router with ID (192.168.30.50) (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 160.1.60.0
            OSPF router with ID (192.5.5.5) (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: 160.1.60.0 (External Network Number )
  Advertising Router: 192.6.6.6                        <<-------------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 192.168.30.70 on Ethernet0 seq 0x20E0 opt 0x2 flag 0x7 len 32 state INIT
OSPF: 2 Way Communication to 192.168.30.70 on Ethernet0,  state 2WAY
OSPF: Neighbor change Event on int Ethernet0
OSPF: DR/BDR election on Ethernet0
OSPF: Elect BDR 192.168.30.70
OSPF: Elect DR 192.168.30.175 DR: 192.168.30.175 (Id) BDR: 192.168.30.70 (Id)
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB17 opt 0x2 flag 0x7 len 32
OSPF: First DBD and we are not SLAVE
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB17 opt 0x2 flag 0x2 len 92 state EXSTART
OSPF: NBR Negotiation Done. We are the MASTER
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB18 opt 0x2 flag 0x3 len 72
OSPF: Database request to 192.168.30.70
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB18 opt 0x2 flag 0x0 len 32 state EXCHANGE
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB19 opt 0x2 flag 0x1 len 32
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB19 opt 0x2 flag 0x0 len 32 state EXCHANGE
OSPF: Exchange Done with 192.168.30.70 on Ethernet0
OSPF: Synchronized with 192.168.30.70 on Ethernet0, state FULL


---snip---                                                      Indicates two interfaces have
OSPF: Nbr 192.168.11.6 has larger interface MTU             <----- different MTU sizes.
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY



-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

```
- Filtering
      + Filtering Specifics
      + Filtering Aggregate
- Conditional Advertisement
- Conditional Route Injection
- Clearing BGP Sessions
- ORF (Outbound Route Filtering)
- BGP Network Migration
      + Local AS
      + Remove Private AS
- Route-maps
- Dampening
- Peer Groups
- Peering Templates
- Regular Expressions
- Fast External Fallover
- Fast Peering session deactivation
- Support for Next-Hop Address Tracking
- Max Prefix
- BGP Policy Accounting
- Troubleshooting BGP
- Output-101
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    The BGP Process
*=====================*
- BGP is a path vector protocol.
- TCP port 179 is used for reliable transport.
- BGP has no periodic updates, it uses triggered updates:
 > Every 5 seconds for internal peers.
 > Every 30 seconds for external peers.
- Periodic keepalives used to verify TCP connectivity:
 > Default every 60 seconds.
- Holdtime interval: Time if passed with no received keepalive, before a notification message is sent, (default = 180 seconds).
- 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 {as-number}                          - Enables BGP routing process
 #bgp router-id {ip}                             - Configures the RID for BGP Process, not used like the IGPs

 #bgp scan-time {scanner-interval}               - Changes the default value of BGP scanner process runs (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 check
                                                        and performing route dampening
```

```
#timers bgp {keepalive} {holdtime}              - Changes the default values (60sec, 180sec) of BGP timers
                                                - 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 in the sending of BGP routing
                                                      updates for a specific neighbor
                                                - If lowered, 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 destination (dst) port 179 or the local router could initiate a TCP session to 1.2.3.4 on dst port 179.
  > By default the source (src) IP is the IP configured on the outgoing interface.
  > This is called the BGP update source, and can be manually configured "neighbor update-source" command.
  > Recursive lookups are used to determine the outgoing interface to the destination.
  > Unexpected BGP session will be refused, which includes the src/dst IP address, dst port, AS-number and authentication.
  > If AS-numbers match between peers, the session according to Cisco IOS is iBGP, else it is eBGP. (Different to vendor 'J')

- The IDLE state indicates that the router is currently not attempting any connection establishments.
- The BGP states are:
  > Idle
  > Active
  > OpenSent
  > OpenConfirm
  > Established

- The BGP Open message contains the following fields:
  > BGP version number   - Has to match between neighbors.
  > Local AS number   - Has to match between neighbors.
  > Holdtime   - Routers agree on lowest suggested value between neighbors.
  > BGP router identifier (RID)
  > Optional parameters

- Test a connection between peers to confirm connectivity, by using "telnet {dst-ip} 179 /source-interface" .

```
-----------
 COMMANDS
-----------
# telnet {peer ip} {port-179} {/source}         - Good for testing connectivity between peers

# debug ip tcp packet detail                    - Good for seeing the TCP session being build, with src and dst IP's and ports
# debug ip tcp transactions                     - Displays all TCP transactions (start of session, session errors, etc.)
# debug ip bgp events                           - Displays the BGP state transitions
# debug ip bgp keepalives                       - Debugs BGP keepalive packets
# debug ip bgp updates  [acl]                   - Displays all incoming or outgoing BGP updates (!!USE WITH CAUTION!!)
# debug ip bgp [ip] updates [acl]               - Displays all BGP updates received from or sent to a BGP neighbor
                                                - [acl] Optionally matching an IP access-list. (Recommended)
#router bgp {asn}
 #neighbor {ip|peer-group} remote-as {asn}      - Defines an external/internal neighbor as per their ASN
 #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 a 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
-----------
 #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 cannot be a number
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    eBGP sessions
*=====================*
```

- Cisco AD (Administrative Distance) for eBGP peers is 20.
- By default the time-to-live (TTL) is set to 1 for eBGP sessions.
- If a eBGP session is configured between two non-directly connected peers, the TTL must be increased with "ebgp multihop" command for the session to come up (This also applies when a loopback interface is used, as a loopback counts as 1 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 behaviour 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, eBGP route based on the AD(20) will
     be chosen as 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
 > To control the max number of parallel internal/external BGP routes that can be installed in a routing table.
 > 2 required conditions:
     > All attributes must be the same, ie weight, local-pref, as-path, origin, med and igp distance.
     > The next hop router for each multipath must be different.

- BGP Dmzlink Bandwidth
 > Used to enable multipath load balancing for external links with unequal bandwidth capacity.
 > To advertise the bandwidth of the link that is used to exit as AS.

COMMANDS

#neighbor {ip|peer-group} ebgp-multihop [ttl]
- By default, eBGP neighbors must be directly connected. (TTL=1)
- This declares a peer to be several hops away. (Specified with TTL)
- Typically used to run eBGP between loopbacks interfaces for load-sharing purposes
- If no TTL entered, the command default 255 is assumed

#neighbor {ip|peer-group} 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 can be listed only on the LEFT

#neighbor {ip} ttl-security hops {hop-count}
- (value from 1-254)
- Lightweight security mechanism to protect eBGP sessions from CPU-based attacks
- Max number of hops that can separate the eBGP peer from the local router

#distance bgp {external ad} {internal ad} {local}
- Sets the AD for eBGP, iBGP, and local routes. Defaults: eBGP-20 & Local/iBGP-200
- This change applies only to routes received after the command has been entered
- {local}: Locally originated routes like aggregates, network command,
      and redistribution

#network {ip/range} 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)

#neighbor {IP} dmzlink-bw
- Used to advertise the bandwidth of the equal links that are used to exit an AS

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   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.
- 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 multiaccess networks.
- Careful with next-hop processing on NBMA networks. The next-hop must be reachable. Rather use a sub-interface interface
      on different subnet or alternatively disable 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 SET-NEXT-HOP
 #set ip next-hop {ip}                           - Changes the next-hop to the IP specified
#router bgp {asn}
 #neighbor {ip|peer-group} route-map {name} {in|out}
                                                 - Applies the route-map to the iBGP peer to change next-hop processing

 #neighbor {ip|peer-group} next-hop-self         - Changes next-hop processing at edge router to the local peering address
                                                 - Instructs iBGP to use this router as the next-hop for routes advertised



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================*
    iBGP Sessions
*====================*
- Cisco AD (Administrative Distance) for eBGP peers are 200.
- Has no next-hop modification by default. Thus a fully meshed iBGP between routers are required for full reachability.
- Because iBGP sessions are usually logical, it is recommended to setup iBGP sessions between loopbacks.
- iBGP loop prevention is done via route suppression/BGP split horizon
 > iBGP learned routes cannot be advertised onto another iBGP neighbors.
 > This rule implies
      >> Fully meshed iBGP peerings (n*(n-1)/2) OR
      >> Route-reflection OR
      >> Confederations


  RR (Route-Reflectors)
*----------------------*
 > Modifies the iBGP split-horizon rule.
 > Depending on the design, actual traffic is not required to go through the RR, only the updates.
 > RR (Route-Reflectors) have different clients:
      >> eBGP neighbors
            => Are normal eBGP neighbors.
            => Received updates will be advertised to other eBGP neighbors, RR clients, and non-clients.
```

```
      >> RR Clients
            => Are 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
            => Are normal iBGP neighbors, (non RR clients).
            => Received updates will be advertised to eBGP neighbors and RR clients.

 > RR Configuration is done only on route reflectors. RR clients use normal peering configuration but only to the RR.
 > Always configure a cluster-ID on route reflectors when the RR's are in redundant clusters.
 > The default value of the cluster-ID if not configured is the BGP router-ID on the route reflector.
       (Not necessary in non-redundant clusters, as the BGP router-ID is unique)

   Confederations
*------------------*
 > Breaks an autonomous system up into smaller confederations/sub autonomous systems.
 > Confederations modify the iBGP next-hop processing rule.
 > It is generally recommended to use private ASN's (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: Confederations
+--------------------------------------------------
|     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-Reflection <<<
 #neighbor {ip-address} route-reflector-client  - Configures an iBGP neighbor to be a client of this route-reflector

 #bgp cluster-id {cluster-id}                    - Optionally assigns a cluster-ID to the route reflector
                                                 - Cluster-ID is a 4-byte value
                                                 - Required only for clusters with redundant reflectors
                                                 - Cluster-ID cannot be changed after the first client is configured

#no router bgp {as-number}                       >>> Confederations <<<
#router bgp {sub-as-number}                      - Removes old BGP process and 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 connected confederation AS's
```

-=-

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

iBGP Synchronization

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

- If an AS is a transit AS, BGP will not advertise a route until all routers in that AS have learned the external route via IGP.
- Legacy rule which is disabled from IOS 12.2(8)T+.
- Designed to prevent black holes when non-bgp routers are in the transit path and don't carry routes about the external next-hop destinations.

- Enable automatic summarization when:
  > Summarization of IGP-to-BGP redistributed routes to major network boundaries are required.
  > Using classful network command to summarize subnets to a major network boundaries.

- Disable automatic summarization when:
  > Summarization on IGP-to-BGP redistribution are not desired.
  > Using classless variant of the network command.

- Solutions:
  > Run BGP on every router in the transit path.
  > Redistribute BGP into IGP.
  > Tunnel BGP over GRE, IPIP, MPLS etc.

COMMANDS

#[no] synchronization	- Disables synchronization between BGP and an IGP - Should be disabled in modern transit AS

-=-

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

Bestpath Selection Process

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

- BGP path attributes
  > Well-Known: 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 a signal to inform that original information may have been lost when the updates were summarized into a single entry.

  > Optional: Is 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 (multi-exit discriminator) (see below)

- Only the best routes(>) are considered candidates to be advertised and candidate to be placed in the routing table.
- A outbound routing policy affects inbound traffic.
- A inbound routing policy affects outbound traffic.
- Prerequisites:
 > A prefix must have IGP next-hop reachability for BGP to consider that route.
 > Synchronization rule must be met or disabled.

- BGP BestPath Selection process 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 a router originated locally.
 4. Prefer the shortest AS paths (only length is compared).
 5. Prefer the lowest origin code (IGP before EGP 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 configured:
      >- When multiple iBGP routes, non-reflected route are preferred above reflected routes.
      >- Then reflected routes with a shorter cluster ID are preferred above routes with longer cluster-lists.
 9. Prefer paths from router with the lower BGP router-ID.

  Attribute : Cisco Weight

- Used for route-selection of OUTBOUND decisions when ONE router has MULTIPLE links to a provider/providers.
- Provides local routing policy, locally significant within a router, and is never routed in updates.
- BGP weights are specified per neighbor with "neighbor weight" command or with a route-map per routes/paths.
- 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 Clearing BGP Sesssion).
- If no weight value is specified, the default value of 0 is applied.
- Routes that the router originates locally have a default value of 32768.
- Routes can be matched on any combination of prefix-lists, AS-path filters, or other BGP attributes.
- Routes not matched by the route-map will be discarded.

 COMMANDS

 #set weight {value}   - Changes the weight in a route-map
#router bgp 1
 #neighbor {ip} weight {weight}   - Sets the weight for all routes received from this neighbor
                                                  - Weight value (1-65535)

 #neighbor {ip} route-map {map-name} in   - Sets the weight only for the routes matched by the route-map for the neighbor

```
  Attribute : Local Preference
*--------------------------------*
- Used for route-selection of OUTBOUND 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, or when doing redistribution,
       or when sending outgoing route updates.
- Default value is 100, the higher value is always preferred.
-----------
 COMMANDS
-----------
 #set local-pref {value}                        - Changes the local preference in a route-map
#router bgp 1
 #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



  Attribute : AS-PATH
*--------------------------*
- Used for route-selection of INBOUND decisions, to decide which return path to use when MULTIPLE pathsd exist.
- AS-path prepending useful in two scenarios:
       > Manipulating the outgoing AS-path length could result in proper return path selection for primary/backup links.
       > Distributing the return traffic load for multi-homed customers.
- To enforce newly set AS-path lenght, re-send BGP updates outbound to the neighbors (refer to Clearing BGP Sesssion)
- 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.

!!> AS-path prepending cannot be monitored or debugged on the sending router. It can only be observed on the receiving router.
-----------
 COMMANDS
-----------
 #set as-path prepend as-number [as-number]     - Prepends an ASN in the route-map
#router bgp 1
 #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 into consideration the length of the AS-path attribute
                                                      RFC 1771 does not include this step



  Attribute : MED
*--------------------------*
- Used for route-selection of INBOUND decision, 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.
- The MED is not propagated outside of a receiving AS.
- A lower MED value is more preferred.
- By default, the MED is considered only during selection of routes from the same AS,
       which does not include intra-confederation autonomous systems.
- Default MED behaviour with redistribution works differently. With the "network" command or redistribution the metric
       in the routing table will be used as the MED.
```

```
-----------
 COMMANDS
-----------
 #set metric {value}                            - Changes the MED in a route-map
#router bgp 1
 #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 AS's
 #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, it is assumed as value 0,
                                                      and thus interpreted as the best metric which is not always wanted.
 #bgp bestpath med confederation                - Allows routers to compare MED's learned from confederation peers
 #bgp deterministic-med                         - Ensures the comparison of MED's 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 are then compared


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
  Attribute : BGP Communities
*==============================*
- OPTIONAL TRANSITIVE, 32-bit number
- 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.
- Routers that do not support communities pass them along unchanged.
- Cisco IOS parser allows the community format as [AS-number]:[low-order-16-bits]


- A 32-bit community value is split into two parts:
 > The high-order 16 bits contains the AS number of the AS which defines the community meaning.
 > The low-order 16 bits have local significance.

- The standards define several filtering-oriented communities:
 > no-advertise : Do not advertise routes to any peer.
 > no-export    : Do not advertise routes to REAL eBGP peers. Will be advertised to intra-confederation peers.
 > local-as     : Do not advertise routes to any eBGP peers. (Either eBGP peers or intra-confederation peers).
 > internet     : Advertise this route to the internet community. Also used to match all communities.


- To enable the NEW format (as:nn) of bgp-communities use the following command.
      # sh ip bgp 6.1.0.0
       community: 6553620
      !
      (config)#ip bgp-community new-format
      !
      # sh ip bgp 6.1.0.0
       community: 3456:210
```

```
- Community values specified with the "set" command in a route-map overwrites existing communities unless the 'additive' keyword
      is specified.
 > Example using a route-map:
      #route-map name
       #match {condition}
       #set community {value} [up to 32 values] [additive]  - Sets the community/ies for matching routes
      !                                                     - [additive] Preserves the original communities and appends new ones
      #router bgp 1
       #neighbor {IP} route-map {map-name} IN|OUT
       #neighbor {IP|Peer} send-community standard          - By default, communities are stripped in outgoing updates

- Community-lists are similar to access-lists, they are evaluated sequentially, line by line.
- All values listed in one line have to match for the line to match and permit or deny a route.

- Standard community-list
  > The keyword 'internet' is used to match any community value.
  > Permit = match, Deny = don't match.
  - Standard community-list
      #ip community-list {1-99} {permit|deny} value [value…]

- Extended community-lists
 > Are like simple community-lists, but allows matching based on regular expressions
 > Use ".*" to match any community value.
  - Extended community-list
      #ip community-list 100-199 permit|deny regexp

- Named community-lists
 > Allows the network operator to assign meaningful names to community-lists.
 > Can be configured with regular expressions and with numbered community-lists.
 > No limitation on the number of community attributes that can be configured for a named community-list.
      #ip extcommunity-list {standard|expanded} {community-list-name} {permt|deny} {community-number | reg-exp}

- 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.
 > Supports iBGP, eBGP multipath load balancing.
 > Indicates the preference of an AS exit link in terms of bandwidth.

-----------
 COMMANDS
-----------
#bgp {ip} dmzlink-bw                          - Distributes traffic proportionally over external links,
                                                    with unequal bandwidth when multipath is enabled
#neighbor {ip} dmzlink-bw                     - Used by BGP to advertise the bandwidth of links which are used to exit an AS
#neighbor {ip} send-community [std|ext|both]  - By default communities are striped in outgoing updates,
                                                    this enables sending communities
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  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.
!! Note !! The default route is not passed through the outbound BGP filters: (prefix-list, filter-list, or route-map)

-----------
 COMMANDS
-----------
 #neighbor {IP} default-originate               - By default, the default route (0/0) is not advertised in outgoing BGP updates.



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================================*
   Originating Prefixes and Filtering
*========================================*
- See conditional-route-injection and conditional-route-advertisement below for additional methods.

   Network Statement
*--------------------------*
 #network {major-network} [mask {net}] [route-map {name}]
                                                - Allows the advertising/originating of major networks into BGP
                                                - If no mask option, a classful subnet would be assumed
                                                - If Auto-Summary is ENABLED:  At least one subnet of the major network is
                                                       required in the RIB before the route is originated in BGP
                                                - If Auto-Summary is DISABLED: A exact route match is required in the RIB before
                                                      the route is originated in BGP
                                                - Route-Map option - Allows network parameters to be modified before they are
                                                       entered into the BGP table
                                                - BGP routes originated through "network" command have a origin code of 'i-igp'

   Redistribution
*--------------------------*
#access-list {number} permit {network}
 #redistribute {igp} [pid] [metric] [route-map] - BGP routes originated through redistribution have a origin code of '?-incomplete'

 #redistribute {static|connected} [route-map] [metric]
                                                - Redistributes local static or connected routes into BGP table


   Aggregation
*--------------------------*
#aggregate-address {aggregate} [mask] [summary-only]
                                                - Specifies a aggregation range in BGP routing process.
                                                - Default = more specific routes are sent
                                                - The aggregate will be announced only if there is at least one network in the
                                                       specified range in the BGP table (not the IGP table)
                                                - Summary-only: Advertises only the aggregate and not the individual networks
                                                - Routes originated with aggregate command has origin code of "i"
```

```
#aggregate-address {aggregate} {mask} suppress-map {route-map} [summary-only]
                                                - Specifies a suppress map to be referenced
                                                - The prefixes 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 routes to unsuppress on a per neighbor basis



   Filtering
*-------------------------*
- AS-path filters are used to selectively choose routes based the the ASN in the AS-path.
 > Incoming routes: The permitted routes are entered the local BGP table, routes that are denied are silently dropped.
 > Outgoing routes: The 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.

- Prefix-filter lists provides greater flexibility than access-lists:
 > By providing the ability to match on subnets.
 > Support for incremental updates.
 > Performance improvement on long filters.


-----------
 COMMANDS
-----------
# sh ip as-path-access-list [filter-list]       - Displays the configured filter lists
# sh ip bgp filter-list {access-list-number}    - Displays all routes permitted by the specified AS-path access-list
# sh ip bgp regexp {expression}                 - Displays all routes matching regular-expression in one or all filter-lists
# sh ip prefix-list {list}[det|sum][longer]     - Displays the prefix-list and the sequence numbers
# sh ip bgp prefix-list {list-name}             - Displays 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-list} [in|out]
                                                - Configures inbound/outbound AS-path filter for specific BGP neighbor

 #neighbor {ip | peer-group} prefix-list list [in|out]
                                                - Applies filters for inbound/outbound BGP routing updates for a neighbor

 #distribute-list prefix-list {prefix-list} {in|out} {routing-process}
                                                - Filters routes redistributed from specified routing process (into|outof) BGP
```

```
CONFIG-SET: 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
|
```

-=-

==

BGP Conditional Route Advertisement

==

- DOC-CD LOCATION
  - > 12.4T Configuration Guides
    - > IP
      - > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
        - > Configuring a Basic BGP Network
          - > Aggregating Route Prefixes Using BGP

- Assume the following topology.

```
        100-------200
           \     /
            \   /
             \ /
             300
              |
          Prefix-A
```

- ASN 300 wants ALL traffic to prefix-A to enter from ASN-200 only, but in the event of link failure between ASN-200 and ASN-300, traffic should be allowed to enter from ASN-100.
- ASN-100 has a weight set, preferring its direct link to ASN-300 for all prefixes.
- Assume ASN-100 is not cooperating in removing the weight value set.

- BGP conditional route advertisement offers an alternative way to affect how traffic enters the AS.
- By conditionally not advertising prefix-A to a ASN-100, ASN-100 is forced to route via ASN-200.
- And in the event of link failure, conditional advertisement will begin advertising prefix-A to the ASN-100.

- By controlling which prefixes get advertised to which neighbors, traffic can be forced to be routed 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:
 > Both of the above prefixes must be 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 ASN-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-MAP 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 advertise-map ADV-MAP non-exist-map WATCH    - Applies conditional route advertisement for AS-100
|
>     #sh ip bgp neighbors 100.1.1.1 | i Condition
>           Condition-map WATCH, Advertise-map ADV-MAP, status: Advertise     - A positive, the WATCH route is down
>     #sh ip bgp neighbors 100.1.1.1 | i Condition
>           Condition-map WATCH, Advertise-map ADV-MAP, status: Withdraw      - A negative, the WATCH route is up
```

```
-----------
 COMMANDS
-----------
# sh ip bgp neighbors {ip}| i Condition         - Shows the condition status of the advertise 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
        > IP
         > 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 a BGP routing table without the corresponding match in the IGP.
- Only prefixes that are equal to or more specific that 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
|      match ip address prefix-list ORIGINATE_ROUTES                        - Specifies 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 which 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 seconds 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.

- Soft Reconfiguration: Outbound or Inbound (IOS 11.2+)
 > Outbound soft reconfiguration resends complete BGP table. It is not configurable and is always enabled.
 > Inbound soft reconfiguration stores a complete BGP table of a neighbor in router memory. (Could be very resource demanding)
 > Inbound soft reconfiguration require configuration to be setup.

- Route Refresh (Soft Reset) (IOS 12.1+)
 > Used to request a neighbor to resend routing info. Useful after config changes to update 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 establishes them again

#router bgp {asn}                               >>> Soft Reconfiguration <<<
 #neighbor {ip} soft-reconfig [inbound]         - This only 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-ip] - Used to recalculate all BGP update-group member sessions
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===================================*
  ORF (Outbound Route Filtering)
*===================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > 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 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 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 configured.

- 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.
- ORF-capable BGP speaker will install ORFs per neighbor.


-----------
 COMMANDS
-----------
# sh ip bgp neighbor                                - Useful dto verify neighbor capabilities
# clear ip bgp {ip} in [prefix-filter]              - Triggers a route refresh from ORF receivers
                                                    - [prefix-filter] option to refresh the remote filter

#router bgp 1
 #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
        > IP
         > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
          > Configuring Advanced BGP Features
           > BGP Network Autonomous System Migration
```

```
 #neighbor {ip} local-as {asn} [no-prepend [replace-as] [dual-as]]]
                                                 - Hide local-ASN feature is useful when necessary to connect to different SP's
                                                        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 removed from the tail(left) only
                                                        of the AS-path before the update is sent
                                                 - Private AS numbers followed by a public AS number are not removed
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  Route-maps for BGP
*=====================*
```

- 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 "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            - Allowes 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 operator 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 if the route-map entry does not contain a match clause.

CONFIG-SET: Route-Map Continue Feature

```
|     route-map MYNAME permit 10
|      match ip add 1
|      set as-path prepend 2001
|      continue 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 ip policy-list {name22}                    - Displays the policy list/s

#ip policy-list {name22} {permit | deny}        - Creates the policy list
#route-map {name} [permit|deny] {seq_no}
  #match policy-list {name22}                   - Configured the route-map to reference the policy-list
  #set {parameter}                              - Executes various set functions
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================*
  BGP Route-Dampening
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
          > Configuring Advanced BGP Features
           > BGP Route Dampening

- Is designed to reduce router processing load caused by unstable routes.
- Defined in RFC 2439.
- Each time an eBGP route flaps, it gets 1000 penalty points (This cannot be configured or changed).
- IGBP 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 a (clear ip bgp *) is regarded as a flap to neighbors, which could cause that path to be suppressed.
- Using a (clear ip bgp * [soft] in) is NOT regarded as a flap to neighbors.

-----------
 COMMANDS
-----------
# sh ip bgp dampened-paths                          - Displays the dampened routes
# sh ip bgp flap-stat [regexp|filter-list|ip]       - Displays 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                            - Displays 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 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
        > IP
         > Cisco IOS IP Routing: BGP Configuration Guide, Release 12.4T
          > Configuring a Basic BGP Network
           > Peer-Groups

- Benefits
 > Reduces the amount of system resources (CPU and memory) necessary in an 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]        - Displays the specified peer group or all peer groups
# sh ip bgp peer-group [peer-group-name] summary- Displays 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]   - Displays info about peer-group update-group calculation,
                                                       the additions and the removals of members
                                                - Displays 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 a BGP parameter specified for the peer group with a neighbor parameter
 #no neighbor {ip} {any-bgp-parameter}          - Removes a BGP parameter specified for the peer group with the neighbor parameter
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Peering Templates
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > 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: Are used to group and apply the configuration of general session commands to groups of neighbors that
      share common session configuration elements.
 > Peer Policy Templates: Are used to group and apply the configuration of commands that are applied within specific NLRI
      configuration mode.


CONFIG-SET: BGP Peer-Templates
+------------------------------------------------------
|     router bgp 6024
|      template peer-policy POLICY                          - Creates a peer policy template, enter policy-template config-mode
|       route-reflector-client                              - Specifies the client a RR-client
|       weight 300                                          - Specifies a weight for all routes from a neighbor
|      exit-peer-policy
|     !
|      template peer-session iBGP                           - Creates a peer session template, enter session-template config-mode
|       remote-as 6024                                      - Configures peering ASN with a remote neighbor
|       update-source Loopback1                             - Use the Loopback interface for sourcing traffic
|      exit-peer-session
|     !
|     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
|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
    Regular Expressions
*============================*
| - Represents 'OR' Statements
      EX: '21|31'       =     Will match either 21 or 31 in a line.

[ ] - SQUARE BRACKET :Represents a range of characters
      EX: [1-4]         =     Will match any in the range 1 to 4.
      EX: [67]          =     Will match either 6 or 7.

. - DOT : Matches any single character
      EX: [1-4].[67]          =     Match 1/2/3/4 then anything character, then 6/7, thus 156 or 397.

^ - CAROT : Matches beginning of string
      EX: ^21 in '213 317 31 218 731'     =     Will only match the first 21.

$ - DOLLAR : Matches end of string
      EX: $31 in '213 317 31 218 731'     =     Will only match the 31 at the end.

_ - UNDERSCORE : Matches any Delimiter (beginning, end, space, tab, comma)
      EX: _31_ in '213 317 31 218 731'    =     Will only match the 31 in the middle.

( ) - PARENTHESIS : are used for "and" operations. To group things together.
      EX: (213|218)_31        =     Matches 213 or 218 followed by 31, ie '213 317' or '218 31'.

{An Atom is a single preceding character or preceding group }
{The special characters *,?,+ all apply repetition to what immediately precedes them}.

* - ASTERISK : Matches ZERO or MORE atoms(single or group of characters)
      EX : _23(_78)*_45_      =     Will match  "23 45"  or  "23 78 45"  OR  "23 78 78 78 78 45".

? - QUESTION MARK : Matches ZERO or ONE atoms
      EX : _23(_78)?_45_      =     Will match  "23 45"  OR  "23 78 45".

+ - PLUS : Matches ONE or more Atoms
      EX : _23(_78)+_45_      =     Will match  "23 78 45"  OR  "23 78 78 78 78 78 78 45".

\ - BACKSLASH : Removes the special meaning of one of the above characters.
      EX: ^\(213_ = will match (213 at the beginning of string.

  REGEX Examples
*-----------------*
_100_             - Passes/passed through AS 100
^100$             - Directly connected to AS 100 (begins and ends in AS 100
_100$             - Originated in AS 100
^100_             - Networks behind AS 100
^[0-9]+$          - AS Paths that is one AS long
^([0-9]+)(_\1)*$  - Networks originating in neighboring AS, with possible prependings
^$                - Networks originating in LOCAL AS
.*                - Matches everything
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
   Fast External Fallover
*============================*
- Fast External Fallover for external peers are 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 is shutdown as soon as a interface reset
      is detected, appose to waiting for the holddown timer (default = 180sec) to expire.
- Disabling BGP fast external fallover, will wait for the holddown timer to expire, before shutting down the neighbor sessions

----------
 COMMANDS
----------
#router bgp {asn}
 #[no] bgp fast-external-fallover         - [Disables] Enables Fast External Fallover globally, thus waits for hold-time to expire

#int s0/0                                 >> Interface Configurationg
 #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, doesnot disable fall-over

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    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

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

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*===========================*
   BGP Fast Peering Session
*===========================*
- 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

- Enable BGP to monitor the peering session of a specified neighbor for adjacency changes and to deactivate the 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 in between 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        - Match any route with a prefix of /28 or more specific
|     !
|     route-map CHECK-NBR permit 10
|      match ip address prefix-list FILTER28                      - Reference the filter
|     !
|     router bgp 45000
|      neighbor 192.168.1.2 remote-as 40000
|      neighbor 192.168.1.2 fall-over route-map CHECK-NBR         - Reset the session if a /28 or more specfic prefix dissappears
|

----------
 COMMANDS
----------
#router bgp {asn}
 #neighbor {IP} fall-over [bfd | route-map]     - Enables BGP fast peering session fall-over


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
  BGP Policy Accounting (PA)
*==============================*
- BGP policy accounting measures and classifies IP traffic that is sent to, or received from, different peers.
- Policy accounting is enabled on a input interface, and counters based on parameters such as community-lists, ASN,
       AS-paths are used to identify the IP traffic.

----------
 COMMANDS
----------
#bgp-policy {accounting|ip-prec-map}            - Accounting: 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                   >>>   {} curl-brackets indicates replaceble values      <<<
*==========================*
*******************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*******************************************************************************************************************************


   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 ASN 100
      >> 2 bytes 0064: The 0064 is the received ASN in HEX, ie 0x0064 = 100 decimal

%BGP-4-MAXPFX: No. of unicast prefix received from ...
%BGP-3-MAXPFXEXCEED: No. of unicast prefix received from ...
      >> When the max-prefix limit from a neighbor is reached


   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 configuration is correct and matching to neighbors configuration?  # sh run | b router bgp

- If eBGP, is the neighbor directly connected? (Should be 1 hop in the trace)       # trace {peer-ip} source {src-int-ip}
 > If not directly connected is multihop configured?                                # sh run | i {peer-ip}.*ebgp-multihop

- Is there IP reachability to neighbor?                                             # ping {peer-ip} source {src-int-ip}
- Are the underlying routing inplace between neighbors?                             # sh ip route {peer-ip}

- IF the obvious check dont help, enable debugging to analyse the TCP session setup # debug ip tcp transactions
 > If the TCP-SYN packet is not answered with a SYN-ACK packet and times out?
      >> Look for ACL's blocking 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 peers "neighbor" statement?       # sh run | i neighbor.*{peer-ip}
      >> If not is the correct source interfaces 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 the AS numbers between the neighbors are correct                   # sh run | i router bgp|remote-as
      >> If using confederations, double check AS numbers                           # sh run | i router bgp|remote-as
      >> Is MD5 password authentication configured correctly?                      # sh run | i neighbor.*password


   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 a exact prefix match in the RIB?      # sh ip route | i {prefix}/{mask}
 > Is there a distribute-list configured blocking prefixes?                    # sh run | i distribute-list

- Is there a aggregate configured that is not advertised?                      # sh run | i aggregate
 > Is there a more specific network 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)
 > Nice debug to see routes entered and removed from the BGP table             # 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}

 > Prefix is selected as best, but not entered into RIB? 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
- Is the expected communities being received? Sending communities enabled?     # sh run | i neighbor.*send-community

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

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
     OUTPUT 101
*=====================*
------->
R1#show 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 high 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
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 mean RIB-Failure.
            - or this could mean 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 ie (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.
```

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

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



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

Overview

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

- RFC-4364, previously RFC-2547, describes a method by which a Service Provider may use an IP backbone to provide IP VPN (Virtual Private Networks) services for its customers. This method uses a highly scalable "peer model", in which the customers' edge routers send their routes to the Service Provider's edge routers. BGP with multiprotocol extensions, are used to exchange the routes of a particular MPLS VPN among the PE routers that are attached to that VPN.

- Traditional IP routing forwards packets based on the destination address.
- MPLS is a forwarding mechanism in which packets are forwarded based on labels.
- 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 - Is a LSR located on the edge of a MPLS network, processes both labeled and unlabeled packets.
 > Ingress E-LSR - Is a router that receives an unlabeled packet and inserts a label (or labels) in front of the IP header.
 > Egress E-LSR - Is a router that receives a labeled packet, removes all the labels, and forwards it on unlabeled.
 > P Router - Provider Router, is a LSR in MPLS VPN terminology.
 > PE Router - Provider Edge Router, is a edge-LSR in MPLS VPNs in MPLS VPN terminology.
 > CE Router - Customer Edge Router, is a client/site router connected to the MPLS domain, but doesn't run LDP.
 > Label - A 4-byte identifier, used by MPLS to make forwarding decisions.
 > Label Binding - Mapping a label to a FEC.
 > FEC - Is a group of packets forwarded in the same manner, over the same path, or with same forwarding treatment.
 > LSP - Label Switch Path, is series of LSR's that forward labeled packets based on a the FEC.
 > PHP - Penultimate-Hop-Popping, is the act of popping a label one hop before the Egress PE.

- MPLS Components:
 > Control Plane
  >> Uses the configured routing protocols to build a routing table, called the RIB (Routing Information Base).
  >> Uses a label exchange protocol to maintain labels internally in a table called the LIB.
  >> Is also responsible for building two tables in the Forwarding Plane, the FIB and the LFIB tables.

 > Data/Forwarding Plane
  >> Consists of two tables, the FIB and LFIB which is responsible for forwarding incoming packets either based
     on IP (unlabeled) or using the label.
  >> Is also responsible to impose/push/insert, swap or dispose/pop/remove labels.

 > RIB (Routing Information Base)
  >> Another name for the traditional IP routing table.
  >> The RIB table structure is : PROTOCOL, PREFIX, NEXT-HOP

 > LIB (Label Information Base)
  >> A label exchange protocol binds labels to networks learned via the routing protocol.
  >> A label exchange protocol stores ALL labels and their bindings/mapping in the LIB table.
  >> The label exchange protocols are LDP, TDP, and RSVP (used by MPLS TE, but it is beyond the scope of the R&S)
  >> LDP is most commonly used, so focus on that. (TDP is old and busted)
  >> A locally significant label is assigned to each IP destination in the RIB, and that binding is stored in the LIB.
  >> Labels are ONLY assigned to non-BGP routes in the RIB table.
  >> The LIB table structure is : NETWORK, LSR/LOCAL, LABEL

 > FIB (Forwarding Information Base)
  >> Is a CEF build table from the information in the RIB table, used for forwarding.
  >> An arriving packet is labeled if a next-hop label is available for the specific destination IP network.
  >> An arriving packet is forwarded unlabeled, if no label for the next-hop exist.
  >> The FIB table structure is : NETWORK, NEXT-HOP, LABEL

```
 > LFIB (Label Forwarding Information Base)
      >> Is a CEF database used to forward labeled packets.
      >> The LFIB table ONLY stores the labels used to forward packets. (Unlike the LIB table that stores ALL label bindings)
      >> Locally assigned labels, that was previously advertised to upstream neighbors, are mapped to next-hop labels,
            previously received from downstream neighbors.
      >> Incoming labels are locally generated labels that is advertised to all adjacent neighbors.
      >> Outgoing labels are the received labels from the adjacent routers.
      >> The LFIB table structure is: INLABEL, OUTLABEL, NEXT-HOP

- Forwarding Combinations:
 > Incoming IP packets
      > Can be forwarded as a IP packets using the FIB.
      > Can be forwarded as a labeled packet using the FIB, after a label was imposed.
      > Will be dropped if the destination is NOT found in the FIB table, even if there is an MPLS LSP to the destination.

 > Incoming Labeled packets
      > Can be forwarded as labeled packets, using the LFIB, after a label was swap, and/or a new label was imposed.
      > Can be forwarded as a IP packet, using the LFIB and FIB, after the incoming label was popped.
      > Will be dropped if the incoming label is NOT found in the LFIB table, even if the IP destination exists in the RIB.

----------
 COMMANDS
----------
# sh mpls interface {int} [detail]              - Displays the MPLS enabled interfaces, their status, and MTU settings
# sh mpls ldp bindings                          - Displays the LIB table (LIB=TIB)
# sh mpls forwarding-table [detail|labels|vrf]  - Displays the LFIB table
# sh ip cef [detail] [summary] [unresolved]     - Displays the FIB table
                                                - [detail]: Displays ingress imposed labels on edge LSR's
                                                - [unresolved]: Shows the unresolved FIB entries

# debug mpls ldp                                - Debugs LDP adjacencies, session establishments, label binding exchanges
# debug mpls lfib                               - Debugs LFIB events: label creations, removals, rewrites
# debug mpls packets [int]                      - Debugs labeled packets switched by router

#mpls ldp router-id {interface} [force]         - Configures the MPLS-ID, (interface must be in a up state to be used)
                                                - Would be the 1st step if done.
                                                - [force]: Forcibly changes the router-id IP address

#ip cef                                 Step1 - Enables CEF switching globally and creates the FIB table. (default=enabled)
#ip route-cache cef                     Step1 - Enables CEF per interface (default=enabled)
#mpls label protocol [ldp|tdp|both]     Step2 - Starts selected label distribution protocol on the specified interface
                                                - From IOS 12.4(3) LDP is default
#mpls ip                                Step3 - Enables label switching (starts LDP) on an interface
                                                - 'tag-switching ip' is the older config version

#mpls mtu {bytes}                               - Configures the MTU size for labeled packets
                                                - 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 interfaces type
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
   Labels
*============================*
- Are 4 byte identifiers used for forwarding decisions.
- Defines the destination and services for a packet, and identifies the forwarding equivalence class (FEC).
- Labels have local significance, because each LSR independently maps a label to an FEC in a label binding.
- Label bindings are exchanged only between adjacent LSRs.

- A FEC (Forwarding Equivalence Class) is a group or flow of packets that are forwarded along the same path or are treated
      the same with regard to the 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 their EXP values might differ.
- The ingress LSR decides which packets belong to which FEC.

- 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.
 > An MPLS QOS FEC is equal to a combination of a destination network and a COS (class-of-service) value.

- At the ingress to the MPLS network, packets are classified and assigned to a specific FEC using a label.
- No further packet classification is done in the MPLS network.
- All packets in an FEC are forwarded using the same next-hop address that assigned that FEC.

- The MPLS Label Structure (Each label is 4-bytes / 32-bits)

      0             19    22  23       31
      +-------------------+-----+---+--------+
      |                   |     |   |        |
      |     L A B E L     | EXP | S |  TTL   |
      |                   |     |   |        |
      +-------------------+-----+---+--------+

 > 20-bit Label                 - Actual label value. (Labels 0 to 15 is reserved)
 > 3-bit Experimental field     - Is used to define a class of service or IP precedence. RFC 5462 renamed this to TC (Traffic Class)
 > 1-bit Bottom-of-Stack        - Indicates if this is the last label in the label-stack. (1=true, 0=false)
 > 8-bit TTL                    - MPLS label TTL, is used to prevent loops.

- The MPLS label is inserted between the Layer2 and Layer3 headers. Often called a SHIM header.
- The router will change the Layer2 frame headers PID (Protocol Identifier) or Ethertype value to indicate that the packet
      is a MPLS labeled packet.

- MPLS uses the following PID values in HEX:
 > Unlabeled Ethernet unicast         - PID = 0x0800
 > Labeled Ethernet unicast           - PID = 0x8847
 > Labeled Ethernet multicast         - PID = 0x8848
 > HDLC Protocol                      - PID = 0x8847
 > PPP Protocol field                 - PID = 0x0281
 > Frame Relay - NLPID                - PID = 0x80
```

- LSRs can perform these operations:
 > Insert (impose) a label on the ingress edge LSR
 > Swap a label
 > Remove (pop) a label on the egress edge LSR

- Label-Stack
 > Is 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 ONLY based on the top label in a stack.

- TDP (Tag Distribution Protocol) uses port UDP/TCP 711.

- LDP (Label Distribution Protocol)
 > LDP uses port UDP/TCP 646.
 > UDP multicast is used to discover adjacent LDP neighbors, while TCP is used to establish a session.
 > An LDP session is established from the router with the higher IP address.
 > LDP sessions should be established between loopback interfaces of adjacent LSRs.
 > LDP Hellos are sent to the multicast address 224.0.0.2.
 > LDP Hellos are sent every 5 seconds.
 > LDP Keepalives are sent every 60 seconds.
 > Be careful of summarizing loopback address. Aggregation breaks an the LSP into two segments+, causing traffic to be dropped.

- Targeted LDP Session
 > Need with nonadjacent LDP neighbors.
 > The UDP hellos are sent to a unicast IP addresses instead of multicast IP addresses.
 > When a neighbor is discovered, the mechanism to establish a session is the same.

- LSP (Label Switch Path)
 > Is a series of LSR's that forward labeled packets based on the FEC.
 > LSP's are unidirectional, return traffic will follow a different LSP.
 > LDP is used to only advertises labels only for the individual segments in the LSP.

- PHP (Penultimate-Hop-Popping)
 > The egress PE router advertises a label value of 3 (IMP-NULL) to the penultimate router, instructing that router to pop the top label, before forwarding the packet onto the egress PE router.
 > PHP removes the requirement for a double lookup to be performed on a egress PE.
 > The LIB table will display a value of imp-null.

- TTL Propagation
 > By default, on MPLS label imposition, the IP TTL is copied to the label TTL for loop prevention.
 > At every hop in the MPLS network, only the top label's TTL is decremented.
 > At MPLS label disposition, the top label's TTL is copied to the IP TTL.
 > Cisco has TTL propagation enabled by default.
 > Disabling TTL propagation can be used to hide P routers in a network. (If disabled, the ingress label TTL is set to 255)
  >> If fully disabled the P router, will not show in a traceroute done from a PE or a CE router.
  >> If only disabled for forwarding traffic, a traceroute from the PE Router will show the P routers and a CE router won't.

```
- Conditional Label Advertising
 > Enables the selective advertisement of only some labels to some LDP neighbors.


CONFIG-SET: Conditional Label Advertising for only loopbacks IPs
+------------------------------------------------------------
|     no mpls ldp advertise-labels                    - Disables default behaviour to advertise all labels
|     access-list 90 permit 192.168.254.0 0.0.0.255   - Matches all loopback addresses
|     access-list 91 permit any                       - Matches any neighbor
|     mpls ldp advertise-labels for 90 to 91          - Labels matching ACL-90 are send to neighbors matching ACL-91
|

----------
 COMMANDS
----------
# sh mpls ldp parameters                        - Displays the LDP parameters on the local router
# sh mpls ldp discovery [vrf|all]               - Displays all discovered LDP neighbors
# sh mpls ldp neighbor [vrf] [int] [detail]     - Displays the individual LDP neighbors

#mpls ldp neighbor [vrf {name}] {ip} targetted  - Establishes a targeted LDP session with nonadjacent neighbor.

#no mpls ip propagate-ttl [forwarded|local]     - Configure IP TTL Propagation
                                                - Be default TTL-Propagation is ENABLED
                                                - Disables TTL-Propagation, useful to hide core routers
                                                - Forwarded: Trace doesn't work for transit traffic labeled by this router
                                                - Local: Trace doesn't work from the router, but transit traffic does

#no mpls ldp advertise-labels                   - Disables default behaviour 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 VPNs
*============================*
- The MPLS VPN architecture enables PE routers to participate in client routing, while maintaining separation between clients and
      optimizing the routing between client sites.
- It also enables separate clients to use overlapping addresses.
- With MPLS VPNs the following tables are duplicated per VPN, the RIB, the FIB, and the LFIB.
- With EIGRP, EBGP, RIP the routing separation is done by several instances in the same process. OSPF implements separate processes.
- Failure to redistribute non-BGP routes into the per-VRF instance of BGP is one of the most common MPLS VPN configuration errors.

- VRF (Virtual Routing and Forwarding)
 > Provides the isolation between different clients running the same internal address space.
 > A VRF consist of a separate RIB and CEF table for each client.
 > A VRF is locally significant to a router.
 > Each interface can only be assigned to one VRF, yet a VRF can have many interfaces assigned.
```

- RD (Route-Distinguisher)
 > Is a 64-bit (8-byte) prepended prefix, used to convert a clients non-unique 32-bit IPv4 address into a unique 96-bit VPNv4 address, to enable transport between PE routers.
 > VPNv4 address are exchanged between PE routers via MP-iBGP.
 > RD uniquely identifies a route (IP prefix), it does NOT identify a VPN.
 > A RD is locally significant to a router.

- RT (Route-Target)
 > Is a 64-bit 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.
 > Export RTs
 >> Identifies the VPN membership, to which the associated VRF belongs to.
 >> Are attached to a client's route, when it is converted into a VPNv4 route.
 > Import RTs
 >> Are used to select which VPNv4 routes are to be inserted into which 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.

- With MPLS VPNs, two labels are used:
 > The outer/top label is used for switching the packet in the MPLS network. (Often called the LDP label)
 > The top label points to the egress router and is propagated by LDP. (Adjacent LSR's label for the next-hop's IPv4 prefix)
 > 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.

- A VPN label is assigned to every VPN route by the Egress PE router, and then advertised to ALL other PE routers in a MP-BGP update. (Don't forget BGP still requires a full-mesh for iBGP)
- The BGP next-hop address must be an IGP route.
- The ingress PE router converts the clients IPv4 routes, exports the VPNv4 routes from VRF tables into MP-BGP and propagates them as VPNv4 routes to other PE routers.
- P routers typically have no knowledge of the VPN routes, as they only swap the LDP labels along a LSP.
- The egress PE router imports the incoming VPNv4 routes from MP-iBGP into the appropriate VRF based on the RTs (Route-Targets) attached to the routes, before passing on the clients IPV4 routes.

CONFIG-SET: Simple Full-Mesh VPN between three sites connecting to three 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 all MPBGP routes with a RT of 123:789
|     !
|     interface Serial2/4
|     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 all MPBGP routes with a RT of 123:789
|     !
|     interface Serial3/2
|     ip vrf forwarding BOB                        - Assigns the interface to VRF-BOB
```

```
|
|PE3#
|     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 all MPBGP routes with a RT of 123:789
|     !
|     interface Serial1/1
|     ip vrf forwarding BOB                           - Assigns the interface to VRF-BOB

----------
 COMMANDS
----------
# sh ip vrf                                     - Displays the list of all VRFs configured in the router
# sh ip vrf [detail] [interfaces] {vrf-name}    - Displays detailed VRF configuration
                                                - Displays interfaces associated with VRFs
# sh ip protocols vrf {name}                    - Displays the routing protocols configured in a VRF
# sh ip route vrf {name} [summary]              - Displays the VRF routing table
                                                - [Summary]: Displays a summary of routes
# sh mpls forwarding vrf {name}                 - Displays labels allocated by an MPLS VPN for routes in the specified VRF
# sh ip cef vrf {name}                          - Displays per-VRF CEF table
# sh ip cef vrf {name} {ip-prefix} {detail}     - Displays details of an individual CEF entry, including label stack


#ip vrf {vrf-name}                        Step1 - 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}                Step2 - This command assigns a route distinguisher to a VRF
                                                - You can use ASN:nn or A.B.C.D:nn format for an RD
 #route-target export {rt}                Step3 - Specifies an RT to be attached to every route exported from this VRF to MP-BGP
 #route-target import {rt}                Step3 - Specifies what MP-BGP route to import into a VRF instance
 #vpn id {oui:vpn-index}                        - (o) Configures a additional VPN identifier for a VRF
#interface fa0/0
 #ip vrf forwarding {vrf-name}            Step4 - This command associates an interface with the specified VRF
                                                - This will clear any existing IP when configured
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=============================*
   Advanced VRF Features
*=============================*
- Route limiting can be done in 2 ways:
 1- Limit the number of routes received from a BGP neighbor.
      >> Restrict the number of routes received from a neighbor.
      >> Default behaviour 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 ...
```

```
 2- Limit the total number of routes in a VRF.
      >> 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.
      >> Default behaviour when the limit is reached, 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....



- VRF Import Filtering
 > Is a route-map configured in a VRF to allow more granular control over the routes imported.
 > A route is imported into the VRF only if at least one RT attached to the route-map matches one RT configured in the VRF and
      the route is accepted (permitted) by the route map.
 > The route-map can match routes using the following criteria:
      >> Access-lists
      >> Prefix-lists
      >> Extended-communities
 > The route-map is configured in addition to a RT import "route-target import {rt}"


CONFIG-SET: MPLS-VPN - VRF Import Filtering example
+-------------------------------------------------------------------
|     access-list 55 permit 10.1.1.0 0.0.0.255        - Matches a /24 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                          - Only matches routes with a RT of 123:2
|     !
|     route-map IMPORT permit 20
|      match ip address 55                            - Only matches the route 10.1.1.0/24
|
|     ip vrf CLIENT-A
|      rd 123:789
|      import map IMPORT                              - Applies the import-map, importing ALL routes with a RT 123:2
|                                                           and 10.1.1.0/24 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-RIB routes with a RT of 123:789



- Selective VRF Export
 > Is a route-map configured in a VRF allowing additional RTs to be attached to the matching routes.
 > You can use a export route-map to filter target routes for a target VPN export by selectively attaching RTs.
 > The route map might deny export to selected routes from a community on the export list.
 > An export-map command with a "set extcommunity rt" command overwrites the configured route targets (RTs), unless
      the additive keyword is specified.
 > Does NOT require a RT export statement, but does filter it "route-target export {rt}".
```

```
 CONFIG-SET: MPLS-VPN - Selective VRF Export
+-------------------------------------------------
|     access-list 55 permit 10.1.1.0 0.0.0.255        - Matches a global route
|     access-list 666 permit 20.1.20.0 0.0.0.255      - Matches a no-export route
|     !
|     route-map EX-MAP permit 10
|      match ip address 55                            - References ACL-55
|      set extcommunity rt 123:555 additive           - Attaches RT 123:55 in addition to RT 123:789 to 10.1.1.0/24
|      !
|     route-map EX-MAP deny 20
|      match ip address 666                           - Prevents 20.1.20.0/24 from being exported
|      !
|     route-map EX-MAP permit 30                      - Required to export other routes with RT of 123:789
|     !
|     !
|     ip vrf CLIENT-B
|      rd 123:789
|      export map EX-MAP                              - Applies the export-map
|      route-target import 123:789                    - Imports all MPBGP routes with a RT of 123:789
|      route-target export 123:789                    - Exports all VRF-RIB routes with a RT of 123:789
|
```

- Hub-Spoke Scenario
 > A hub-spoke design could be used when full connectivity between sites are 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 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.
+-------------------------------------------------------------
| In this example you have 3 client sites, connecting via one HUB-site to each other.
| To illustrate a catch, Site-1 and Site-2 connects to the same PE2 router.
| The HUB-site connect 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                           - Assign 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                                 else the Site-1 and Site-2 will share VRF-BOB, and
|     rd 123:22                                          thus be allowed to communicate directly
|     route-target export 123:200                   - Export 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

----------
 COMMANDS
----------
#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 is 75%)
                                                  - [Warning-only]: Warning when exceeding appose to dropping the session
                                                  - [Restart] : Re-establish the dropped session after the time specified

#ip vrf {name}
 #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

#ip extcommunity-list {no} {permit|deny} rt {no}- Creates a community-list

#route-map {name} {permit|deny} [seq]
 #match ....                                      - Matches the necessary
 #set extcommunity rt {value} [additive]          - Used to attached additional RTs in export-maps
                                                  - [additive] will append this RT and not overwrite original's set

#ip vrf {name}
 #import map {route-map}                          - Configures selective VRF import
 #export map {route-map}                          - Configures selective VRF export
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
   PE and PE: MP-iBGP
*==============================*
- BGP on a MPLS VPN enabled router utilize separate contexts (configured as address-families):
 > Global BGP routing (IPv4) is commonly used to propagate internet routes.              -> (address-family ipv4)
 > MP-iBGP is used for BGP communication between PE routers to propagate VPNv4 routes.    -> (address-family vpnv4)
 > MP-eBGP is used for BGP communication between CE and PE routers.                       -> (address-family ipv4 vrf)

- Do not forget to activate neighbors under the address-families.
- By default, the exchange of IPv4 routes between MP-iBGP neighbors is enabled.
- This can be disabled in 2 ways:
 > For all IPv4 routes        : no bgp default ipv4-unicast
 > For specific neighbor      : no neighbor activate (within address-family ipv4)


CONFIG-SET : MP-BGP, Limit route-exchange for neighbors to specific address-families
+-------------------------------------------------------------------
|     router bgp 65000
|      neighbor 10.1.0.1 remote-as 65000        - Specifies the BGP neighbors
|      neighbor 10.1.0.5 remote-as 65000
|      neighbor 10.1.0.9 remote-as 65000
|      !
|      no bgp default ipv4-unicast              - Disables default IPv4 route exchange for all neighbors
|      neighbor 10.1.0.1 activate               - Manually enables IPv4 route exchange for these two neighbors
|      neighbor 10.1.0.5 activate                    only because default ipv4-unicast was disabled
|      !
|      address-family vpnv4                     - Enter PE-PE MP-BGP configuration context
|       neighbor 10.1.0.5 activate              - Manually enable VPN-v4 routes for these two neighbors
|       neighbor 10.1.0.9 activate

----------
 COMMANDS
----------
# sh ip bgp vpnv4 [all|rd|vrf{name}] labels     - Displays the labels associated with VPNv4 routes
# sh ip bgp vpnv4 vrf {name}                    - Displays per-VRF BGP parameters
# sh ip bgp vpnv4 rd {asn:nn}                   - Displays the NLRI prefixes that have a matching RD
# sh ip bgp vpnv4 all                           - Displays whole VPNv4 table
# sh ip bgp neighbors [ip]                      - Displays global BGP neighbors and the protocols negotiated with these neighbors

#router bgp {asn}
 #neighbor {ip} remote-as {r-asn}               - All MP-BGP neighbors have to be configured under global BGP routing config
 #neighbor {ip} update-source loopback0         - MP-iBGP sessions should run between loopback interfaces
 #address-family vpnv4                          - Selects configuration of VPNv4 prefix exchanges under MP-iBGP sessions
  #neighbor {ip} activate                       - The BGP neighbor defined under BGP router configuration has to be activated
                                                     for VPNv4 route exchange
  #neighbor {ip} [next-hop-self]                - Changes the 'next-hop' ip to local peer address
  #neighbor {ip} send-community [std|ext|both]  - Extended community is required for RT propagation

 #no bgp default ipv4-unicast                   - Disables the default exchange of IPv4 routes for all neighbors
                                                - Neighbors that need to receive IPv4 routes needs to be activated
 #no neighbor {ip} activate                     - Disables IPv4 route exchange on a per neighbor basis
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==============================*
   PE to CE: Static route
*==============================*
- Don't forget to redistribute the static routes into MP-BGP on the ingress PE router.
- You must specify a next-hop IP if you are using a non-point-to-point interface.

----------
 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 routing table
                                                - [permanent]: Route stays in the RIB even if interface is shut down



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
   PE to CE: RIP
*========================*
- Only RIPv2 is supported for PE-CE routing!
- Don't forget to redistribute the RIP routes into MP-BGP on the ingress PE router, and back into RIP 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.
 > On the egress PE router, the RIP hop count has to be manually set for routes redistributed back into RIP. Alternatively the
      'metric transparent' option can be used to copy the BGP MED into the RIP hop count, for a consistent end-to-end RIP hop count.

----------
 COMMANDS
----------
# sh ip route vrf {name} rip                    - Displays the RIP routes in the VRF-RIB
#router rip
 #version 2                                     - Version 2 must be used
 #address-family ipv4 vrf {name}                - Creates a per-vrf context within RIP routing process
  #redistribute bgp {asn} metric 5              - Redistributes BGP routes. 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
*=========================*
- Don't forget to redistribute the EIGRP routes into MP-BGP on the ingress PE router, and back into EIGRP on the egress PE router.
- With EIGRP you must specify the AS-number within the per-vrf context.
- The IGP metric of a route is always copied into the BGP MED attribute when redistributed into BGP.
- The BGP MED is NOT copied back into the IGP metric.
- For external EIGRP routes (redistributed routes from BGP) and non-EIGRP routes, the metric must be configured, before a route
      is redistributed into an EIGRP-CE router.
- External routes received without the configured metric will not be advertised to the CE router.

- EIGRP SOO Loop-Prevention
 > The SSO (Site Of Origin) extended community can be used to prevent loops in dual-homed scenarios or when a
      backdoor is configured.
 > A unique SOO value must be configured for each VPN site.
 > When a router receives a route on a interface with a sitemap configured and the SOO of the route matches the configured SOO,
      the route is rejected.
 > This value must be used on the PE-CE interface.

----------
 COMMANDS
----------
# sh ip route vrf {name} eigrp                  - Displays the EIGRP routes in the VRF-RIB

#router eigrp {pid}
 #address-family ipv4 vrf {name}                - Creates a per-vrf context within EIGRP routing process
 #autonomous-system {asn}                       - Configures per-vrf AS number
 #redistribute bgp {asn} metric {b d l r m}     - Redistributes 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 inbound routing updates
                                                      received from this interface
```

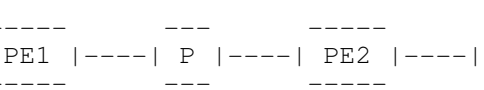
```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================*
   PE to CE: OSPF
*==========================*
- Don't forget to redistribute the OSPF routes into MP-BGP on the ingress PE router, and back into OSPF on the egress PE router.
- OSPF is not fully VPN-aware and requires a separate OSPF process when configured.
- OSPF had to be extended to keep most of the functionality had in vanilla OSPF.
- A new extended BGP community was defined to carry OSPF route types and area types across the BGP backbone.
- PE routers are seen as ABRs (Area Border Routers).
- Routes redistributed from BGP into OSPF appear as inter-area summary routes or as external routes (based on their
      original LSA type) in other areas.


- MP-BGP to OSPF redistribution rules:
 > For original Type-1 or Type-2 LSA's, the redistributed routes will appear as inter-area summary LSA (Type-3).
 > For original Type-3 LSA's, the redistributed routes will appear as inter-area summary LSA (Type-3).
 > For Type-5 LSA's, the LSA's are reoriginated as Type-5 LSA's with the egress PE as a ASBR.
 > For Type-7 LSA's the LSA's are announced as Type-5 LSA's, 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 Down-Bit
 > An additional bit was been introduced in the options field of the OSPF LSA header, to prevent loops.
 > PE routers set the down bit when redistributing routes from MP-BGP into OSPF.
 > PE routers will never redistribute OSPF routes into MP-BGP, if the down-bit is set.


- Sham-Link
 > Scenario: 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 in each PE router for each sham-link.
 > This /32 must always be advertised by MP-iBGP, not by OSPF, and must belong to the VRF.


----------
 COMMANDS
----------
# sh ip route vrf {name} ospf                     - Displays the OSPF routes in the VRF-RIB

#router ospf {pid} vrf {name}                     - Starts a separate OSPF routing process for every VRF
 #redistribute bgp {asn} subnets                  - Redistributes MP-BGP routes into OSPF
                                                  - The subnets keyword is needed to avoid classful routes
 #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
*========================*
- There is not a separate per-VRF BGP table, there is only a global MP-BGP table.
- To configure eBGP as a PE-CE routing protocol, just configure the CE router as a eBGP neighbor under the ipv4 address-family.
- Don't forget to activate the neighbor.

- AS-Override
 > Default BGP loop prevention rules disallow discontiguous autonomous systems.
 > It allows the same AS-number to be used on different sites, by ignoring the AS-path loop prevention rule.
 > It is required when a client has two sites with the same AS-number interconnected by MP-BGP.
 > If the first AS number in the AS path is equal to the neighboring AS,it is replaced with the provider AS number.
 > Prepending is allowed. All AS-number occurrences are replaced with the provider AS number.

- Allowas-in
 > By default, a BGP router cannot accept a prefix if the local ASN is listed in the received as-path list.
 > This default behaviour can be changed with the 'neighbor allowas-in' command.

- SOO (Site-of-Origin)
 > Attribute is used to prevent PE-CE-PE and CE-PE-CE routing loops in multihomed environments.
 > A route inserted into a VRF is not propagated to a CE router if the SOO attached to that route is equal to the SOO attribute
      associated with the CE router.


----------
 COMMANDS
----------
# sh ip route vrf {name} bgp                 - Displays the eBGP routes in the VRF-RIB
# sh ip bgp vpnv4 vrf {name}                 - Displays the eBGP routes in the BGP table
# sh ip bgp vpnv4 vrf {name} summary         - Displays the VRF neighbors


#router bgp {asn}
 #address-family ipv4 vrf {name}             - Select MP-BGP VRF- context
 #neighbor {ip|peer-group} remote-as {asn}   - Configures eBGP neighbor in the VRF context, not in the global BGP config
 #neighbor {ip|peer-group} activate          - CE neighbors must to be activated
 #neighbor {ip|peer-group} as-override       - Configures AS-Override
 #neighbor {ip|peer-group} allowas-in {no}   - Disables the default EBGP loop-prevention, allowing the local ASN 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)
*===========================*
- VRF-lite allows the CE router the ability to maintain separate VRF tables, with the purpose to extend the privacy and security
      of an MPLS VPN down to a branch office interfaces.
- The CE router separates traffic between client networks using VRFs.
- 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.

- IF the Multi-VRF CE router runs OSPF, you must configure the 'capability vrf-lite' under the OSPF process.
 > OSPF multi-VRF allows you to split the router into multiple virtual routers, where each router contains its own set of
      interfaces, routing table, and forwarding table.
 > OSPF multi-VRF gives you the ability to segment parts of your network and configure those segments to perform specific
      functions, while still maintaining correct routing information.


CONFIG-SET : VRF-lite CE configuration example, (PE config remains unchanged)
+------------------------------------------------------------------------------
|     ip vrf LEET
|      description Trusted Traffic
|     ip vrf NOOB
|      description Guest Traffic
|     !
|     interface FastEthernet2/0.10
|      encapsulation dot1Q 10
|      ip vrf forwarding LEET                - Places the interface into the LEET VRF
|      ip address 10.0.12.1 255.255.255.252
|     !
|     interface FastEthernet2/0.20
|      encapsulation dot1Q 20
|      ip vrf forwarding NOOB                - Places the interface into the NOOB VRF
|      ip address 192.168.12.1 255.255.255.252
|     !
|     router ospf 1 vrf LEET
|      router-id 0.0.1.1
|       network 10.0.0.0 0.0.255.255 area 0
|       capability vrf-lite                  - Enables onwards OSPF advertisements
|
|     router ospf 2 vrf SCUM
|       router-id 0.0.1.2
|       network 192.168.0.0 0.0.255.255 area 0
|       capability vrf-lite                  - Enables onwards OSPF advertisements
|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   Troubleshooting MPLS             >>>   {} curl-brackets indicates replaceble values      <<<
*=========================*
*********************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*********************************************************************************************************************************

   Troubleshooting MPLS and LDP
*---------------------------------*
- LDP Session Startup Issues
 > Does all the expected neighbors show up?                               - #sh mpls ldp discovery
 > Is MPLS enable on all the necessary interfaces?                        - #sh mpls interface
 > Is all expected neighbors directly adjacent?                           - #trace {neighbor} (should be 1 hop)
 > Are all neighbors using the same protocol: LDP/TDP?                    - #sh mpls interface detail
 > Any interface ACLs dropping ports 711 or 646?                          - #sh run int {interface}
 > Test connectivity between loopback interfaces                          - #ping {ip} source {ip}'

- Label 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 adjacent LSR display received labels?                             - #sh mpls ldp bindings (on neighbor)
 > Is conditional label advertising configured?                           - #sh run | i advert

- Problem with large packets
 > Does a extended ping with packet sizes close to 1500 fail?             - #ping {IP} size 1500 df
 > Is the correct MTU's set?                                              - #sh mpls interfaces detail | i MTU
```

Troubleshooting MPLS VPN's



- Verifying proper routing information flow end-to-end (left-to-right)
 > Are the CE routes received by a ingress PE1 router? - PE1# sh ip route vrf {NAME}

 > Routes redistributed into MP-BGP have proper extended communities? - PE1# sh ip bgp vpnv4 vrf {NAME} {prefix}
 > Have the PE routers exchange their allocated VPN labels? - PE1# sh ip bgp vpnv4 all labels

 > 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}

 > On PE1 does the BGP nexthop prefix have LDP label received from P? - PE1# sh mpls forwarding-table {prefix}
 > Is there a end-to-end (PE1-to-PE2) LSP? Verify on PE1, P and PE2 - ALL#sh mpls forwarding-table | in ^locallabel

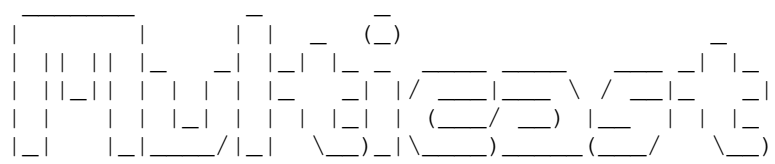
 > Is the BGP route selection process working correctly? - PE2# sh ip bgp vpnv4 vrf {NAME} {prefix}
 > Is the expected best routes installed into the VRF-RIB on PE2? - PE2# sh ip route vrf {NAME} {prefix}

 > If multiple equal cost links links(CE1=PE1):
  -> Is only one route installed in to VRF-RIB on PE2? - PE2# sh ip route vrf {NAME} {prefix}
  -> Is BGP multipath enabled? - PE2# sh run | i maximum-paths.*address-fa

 > 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}

- Identifying the issues when verifying the data flow
 > Is CEF enabled on the ingress PE1 router interface? - PE1# sh cef interface
 > Is the correct labels allocated on the ingress PE1 router?
  -> Does the top-label (LDP) correspond to the BGP next-hop label (P-route)? - PE1# sh ip cef vrf {NAME} {prefix} detail
  -> Does the second-label (VPN) correspond to the VPN label from PE2? - PE2# sh mpls forwarding-table vrf {NAME}

THIS PAGE WAS LEFT BLANK INTENTIONALLY



-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

```
- Filtering
      + Static RP
            o Filtering Specific Groups
      + Auto-RP filtering
            o RP group filtering
            o MA filtering RP's
      + BSR
            o Specific Groups
      + PIM-Neighbor filtering
      + Client filtering
      + Multicast Boundary
      + Multicast Route-Limit
- Scoping
      + TTL Scoping
      + Administrative Scoping
- Additional Multicast features
      + Multicast Rate Limiting
      + 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)
- NTP via Multicast
- Troubleshooting Multicast


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=========================*
  Multicast Operation
*=========================*
- Multicast is UDP-based, and thus is unreliable by design.
- A multicast server is usually the source, client is 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 the routers
#ip multicast-routing distributed             - Globally enables multicast routing on the 3560
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*======================*
    Addressing
*======================*
- Multicast-address class-D range : (224.0.0.0-239.255.255.255),  224.0.0.0/4,   224.0.0.0/240.255.255.255,   224.0.0.0/15.0.0.0

 > Reserved Link Local addresses                           224.0.0.0/24
      >> These are non-routed addresses used only on a local link. (TTL=1)

 > Reserved Local Routed addresses                         224.0.1.0/24
      >> Reserved for network protocols which needs to be forwarded throughout a network.

 > Reserved Source-Specific Multicast (SSM) range          232.0.0.0/8
      >> Allows IGMPv3 hosts application to select a source for the 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 then 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           - Cisco Auto-RP Announce (RP)
 > 224.0.1.40           - Cisco 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
                   \   / \ /  \ /  \ /  \ /  \ /
                    \ /   v    v    v    v    v
         01-00-5E-   4    D -  6    2 -  B    1

      Combine the last output to get 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

- IGMP v1 and v2 uses protocol number 2.
- IP TTL=1.
- Enabled automatically with PIM.
- Designed to enable communication between a multicast router and connected hosts.
- By informing the local multicast router a host wants to receive traffic for a specific group.
- Or to inform the router that a host 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 hosts NIC will start listening for that multicast MAC address too.

- IGMP v2 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:
  >> Allows 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 to query messages on a segment with multiple routers.
  >> The router with the lowest IP address is elected as the IGMP querier.

- 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 which 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 Timers
 > Query interval
    >> A time period between general queries sent by a router.
 > Query response interval
    >> Max response time for hosts to respond to the periodic general queries.
 > Group membership interval
    >> A time period during 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
    >> A time period during 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,
    through a feature called Source-Specific Multicast (SSM).
 > It allows a host to indicate interest in receiving packets only from specific source addresses, or from all but specific
    source addresses, sent to a particular multicast address.
 > Leave group messages are sent to destination 224.0.0.22.

- IGMP Snooping
 > IGMP snooping enables the switch software to eavesdrop on the IGMP conversation between multicast hosts and the router.
     The switch examines IGMP messages and learns the location of multicast routers and group members.

- IGMP testing and verifying commands, does not enable IGMP, (IGMP enable with PIM).

 COMMANDS

```
#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 to join the destination mgroup
                                               - Interface will process multicast traffic, so will answer pings
#ip igmp static-group {m-ip}                   - Only emulates the join, but doesn't process multicast traffic
                                               - Interface will not process multicast traffic like pings
#ping {m-ip}                                   - Emulates a server multicasting to a group
```

-=-

=====================
PIM
=====================

- DOC-CD LOCATION
  > 12.4T Configuration Guides
  > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
  > Configuring Basic IP Multicast

- IS used for router to router communication. PIM is a multicast routing protocol.
- Does not advertise its own topology information, it relies on unicast routing protocols.
- Multicast groups can only be 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 received by.

- Designated Router
  > A DR is necessary for each multi-access LAN which runs 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.

- 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 a tie, then the router with highest IP address will elected.

Dense Mode (PIM-DM)

> Is designed for networks which have many multicast clients which are tightly spaced together.
> When a PIM-DM router receives a multicast packet, it first performs the 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 one on which it received the packet and on interfaces 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 the Shortest-Path-Tree(SPT) may differ for each combination source and multicast group.
> The notation (S,G) refers to a particular dense SPT.
> The SPT includes all interfaces by default, but PIM Prune message allows interfaces to be removed.

> Dense-mode routing protocols include DVMRP, MOSPF, and PIM-DM.
> Implicit join assumes all traffic is wanted by all clients, unless client specifies they don't want it.
> Flood and Prune behaviour
  >> 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 outgoing interface list is indicates as NULL.

> A multicast router can have more than one interface in the outgoing interface list, but it
    can have only one interface 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 the source is still unknown.
> Upstream is towards the source, and downstream is towards the multicast group/hosts.

> A router sends a graft message to an upstream neighbor, a neighbor to which it had formerly sent a prune message,
    causing the upstream router to put the link back into a forwarding state (for a particular (S,G) SPT).

Sparse mode (PIM-SM)

> Designed for only a couple clients in a large network.
> Sparse-mode protocols do not forward group traffic out of any interface, unless a router requests copies of packets sent
    to a particular multicast group. A downstream router will request this either because another downstream router
    requested it, or a host directly connected sent an IGMP join message for that group.
> Explicit join: no traffic is sent unless asked for.
> Sparse-mode employs Rendezvous Points (RP) to process join requests.

> Order of operation with sparse:
    >> The multicast source begins sending multicast traffic to the group, eg: 226.1.1.1
    >> The connected router sends a unicast PIM Register messages to the defined RP (10.1.1.1,226.1.1.1)
    >> Transit PIM Routers WONT show this (S,G) since it was a unicast PIM message.
    >> The RP will ignore this multicast traffic until it receive PIM Join for that group,
          and will reply with a register stop messages: "I don't have any sources, stop sending traffic".
    >> A host somewhere send a IGMP JOIN (*,226.1.1.1) to its gateway router.
    >> This downstream gateway router then send a PIM-SM JOIN for (*,226.1.1.1) to the defined RP.
    >> PIM routers in transit path will install (*,226.1.1.1) into their multicast tables.
    >> The RP then starts 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 (10.1.1.1,226.1.1.1) to the RP.

> PIM-SM typically uses a Shared-Path-Tree or Root-Path-Tree(RPT), because it is rooted at the RP.
> PIM-SM initially causes multicasts to be delivered in a two-step process: first, packets are sent from the source to
    the RP, and then the RP forwards the packets to the subnets which have hosts that need a copy of those multicasts.
    PIM-SM uses a shared tree in 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.

Tree-Types

- 2 Types:
> Source-Tree or Shortest-Path-Tree(SPT) 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 also the default type.
> Shared-Tree or Root-Path-Tree(RPT), 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 source-tree (SPT),
!!!- Changing the tree-type from a source-based to a shared-tree could be used as a workaround with an RPF failure,
      specifically when the use of static mroutes, and changing of the unicast routing is not allowed.


 Shortest Path Switchover
----------------------------
 > Is 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 Shortest-Path-Tree(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
  “ip pim spt-threshold” command.

-----------
 COMMANDS
-----------
# sh ip pim neighbors                              - Shows PIM neighbors

 #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 or ACL groups
                                                 - {kbps} Traffic rate in kilobits per second before switchover
                                                 - {infinity} Never switch to source-tree


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==================================*
    Reverse Path Forwarding (RPF)
*==================================*
- PIM relies on unicast routing protocols for the Reverse Path Forwarding (RPF) check.
- Cisco defining the RFP Check as follow:
 > When a router receives a multicast packet, the source IP is taken, and used to determine the reverse path interface,
      by looking at the unicast routing table to determine the interface used to forward traffic back to this
      source. If this reverse path interface matches the interface the multicast traffic was received on,
       the RPF check is successful. If not the packet is ignored.

- The RPF basically check verifies that the incoming interface for multicast feed is outgoing interface for unicast traffic
       back towards the source.

- Static "mroute" overrides unicast information, by allowing additional interfaces to receive mtraffic.
- Static "mroute" has no influence on data flow. Only used for the RFP check.

- TO verify RPF failure:
 > show ip mroute count - "RPF failed" counter would indicate the amount of RFP failures seen.
 > debug ip mpacket     - "not RPF interface" is bad. Points out a RPF failure.
                        - "mforward" is good.
                        - Must disable "ip mroute-cache" to debug transit traffic.
```

```
-----------
 COMMANDS
-----------
# 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 lack of client interest
# show ip rpf {ip}                            - Shows the RPF information regarding a IP

# debug ip mpacket                            - Shows the multicast packet information
                                              - Requires "ip mroute-cache" to be disabled before
# debug ip pim                                - Shows the PIM events and transactions

#ip mroute {source ip} {mask} {nh}            - Changes the interfaces for which a 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:
 > Statically assigned with "ip pim rp-address {IP}"
      >> Static RP Assignments by default are LESS preferred than dynamically learned RPs.
 > Dynamically via
      >> Cisco proprietary Auto-RP or
      >> Standards based Bootstrap Router (BSR)

- To use redundant RPs, Cisco offers two methods:
 > Anycast RP using the Multicast Source Discovery Protocol (MSDP)
 > Bootstrap Router (BSR)

  Cisco's 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

- 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 candidate-RPs
      by listening to the RP-announce messages.
 > The mapping agent builds a mapping table which lists the current RP's for each range of multicast groups, then the
      mapping agent picks the RP with the highest IP address if multiple RP's support the same multicast groups.
 > The mapping agent sends RP-discover messages (224.0.1.40) advertising the mappings and RP's to other routers.
 > All mrouters 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:
      >> Candidate RP (224.0.1.39):       #ip pim send-rp-announce
      >> Mapping Agent(224.0.1.40):       #ip pim send-rp-discovery

- When using a loopback interface for the discovery or candidate-RP advertisement, ensure to enable PIM on that interface.
- A design problem with auto-RP, is it was designed for sparse but to find the mapping, you need dense behaviour.
- The problem with auto-RP router, in sparse-mode is:
      >> Can't join the auto-RP groups without knowing where the RP is.
      >> Don't know where the RP is without joining the auto-RP groups.
- The c-RP announcements and mapping agent discovery requires dense mode, so the two workarounds are:
      >> Sparse-Dense
            + Dense for groups without RP (including: 224.0.1.39/224.0.1.40)
            + Sparse for all other
      >> Auto-RP Listener
            + ONLY 224.0.1.39 and 224.0.1.40 to run in dense mode,
            + Sparse for others
            + Does not require the interface to be in dense mode
            + Command "ip autorp listener" on some older IOS's are hidden
            + Configure on all multicast routers

- Sparse-Dense Mode
 > To get around the chicken-egg problem mentioned above with auto-RP and PIM-SM, Cisco created sparse-dense-mode.
 > In PIM sparse-dense mode, a router uses PIM-DM when it does not know the location of the RP, and PIM-SM
      when it does know the location of the RP. So, under normal conditions with auto-RP, the routers would use dense
      mode long enough to learn the group-to-RP mappings from the mapping agent, and then switch over to sparse mode.
 > Every group for which the RP is unknown, the tree will fall-back to dense-mode.

 > Sparse for groups with an RP, dense for all others which don't have a RP.
 > The newer workaround instead of using spare-dense:
      #ip pim autorp listener             - Typically used when there are only sparse-mode interfaces configured


   Bootstrap Router (BSR)
*-------------------------*
- 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.
- Is often referred to as PIMv2. It is common being asks to configure PIMv2, implying BSR.
- One router acts as BSR, which is similar to the mapping agent in auto-RP.
- The BSR receives mapping information from the RPs, and then it advertises the information to other routers.
- However, some differences between the BSR, and the auto-RP mapping agent:
 > 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 PIM routers inside bootstrap messages.
 > PIM routers each independently pick the currently best RP for each multicast group by running the same hash algorithm
       on the information in the bootstrap message.
 > The BSR floods the mapping information in a bootstrap message sent to the all-PIM-routers multicast address (224.0.0.13).
 > The flooding of the bootstrap message does not require the routers to have a known RP or to support dense mode.
```

- PIM-SM routers flood bootstrap messages out 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.
- c-RP can make use of a priority, (highest wins and def=0) it gives preference to one RP over another.
- When multiple c-RP are advertising the same group-addresses, Cisco IOS will make its decision process on the c-RP that advertises the longest match in the announced groups, then only the RP-priority.
- The BSR feature supports redundant RPs and redundant BSRs.
- Multiple BSR routers can be configured. In that case, 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. Then, the winning BSR, called the preferred BSR, continues to send bootstrap messages, while the other BSRs monitor those messages. If the preferred BSR's bootstrap messages cease, 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 relates 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 IGP's.
- At the end of the process, any packets sent to the RP are routed per the IGP routes, to the closest RP.
- The two biggest benefits of this design 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

```
# sh ip pim interface                              - Shows the interfaces with PIM configured
                                                   - Displays 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 defines the RP on all routers including the RP
                                                   - [acl]: Limits the groups a RP will advertise via PIM-JOIN
                                                   - [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 c-RP
                                                   - {ttl}: The scope ttl of the advertisement message
                                                   - {acl}: See filtering section below
                                                   - {interval}: How often the candidate announcements are sent
```

```
#ip pim send-rp-discovery {src-int} scope {ttl} - Defines the mapping agent
                                                - {int}: The IP address to advertise as the mapping agent
                                                - {ttl}: Is the scope ttl of the discovery message

#interface fa0/0
 #ip pim sparse-dense-mode                      - Uses PIM-SM if RP known else PIM-DM is used
                                                - Alternative to this is "ip autorp listener"

#ip pim autorp listener                         - Needed on ALL routers where auto-RP announcements transits sparse only interfaces
                                                - 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}]
                                                - Configured the BSR candidate-RP

#no ip pim dm-fallback                          - Disables tree to fall-back to dense-mode if RP is unknown
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  Bi-directional PIM
*=====================*
```

Bi-directional PIM

- 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 works efficiently with a relatively small number of multicast senders.
- However, in cases with a large number of senders and receivers, PIM-SM becomes less efficient.
- Bidirectional PIM addresses this inefficiency by slightly changing the rules used by PIM-SM.

- Similar to PM-SM, bidirectional PIM, follows these steps:

1. As with normal PIM-SM, the RP builds a shared tree, with itself as the root, for forwarding multicast packets.
2. When a source sends multicasts, the router receiving those multicasts does not use a PIM Register message. Instead, it forwards the packets in the opposite direction of the shared tree, back up the tree toward the RP. This process continues for all multicast packets from the source.
3. The RP forwards the multicasts via the shared tree.
4. All packets are forwarded per Steps 2 and 3. The RP does not join the source tree for the source, and the leaf routers do not join the SPT either.

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   NBMA Mode
*=====================*
- 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 NMBA to allow traffic to enter and exit the same interface.


-----------
 COMMANDS
-----------
#int s0/0
 #ip pim spare-mode                              - NMBA mode only works with sparse interfaces
 #ip pim sparse-dense mode                       - This command will produce the error, just ignored it
 #ip pim nbma-mode                               - Enables NMBA mode on the interface



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  Multicast over GRE
*=====================*
- 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 "keepalive" under tunnel interface.
- Make sure the source and destination IP’s are not routed through the tunnel interface.
- Unless unicast source is reachable out the tunnel, a RPF failure will occur, which can be fixed with an mroute.
- PIM can be tunnelled inside GRE to transport multicast over unicast-only networks with:


-----------
 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
```

```
*-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-*
*=======================*
  MBGP (Multicast BGP)
*=======================*

 COMMANDS
-----------
# sh ip bgp ipv4 multicast                      - Displays the learned multicast routes

#router bgp 1
 #neighbor 120.1.12.2 remote-as 2               - Specifies the remote ASN peer
 !
 #address-family ipv4
  #neighbor 120.1.12.2 activate                 - Activates the neighbor for unicast traffic
  #network 10.5.5.0 mask 255.255.255.0          - Originates the multicast source network
 !
 #address-family ipv4 multicast
  #neighbor 120.1.12.2 activate                 - Activates the neighbor for multicast traffic
  #network 10.5.5.0 mask 255.255.255.0          - Originates the multicast source network



*-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-*
*=========================*
  Multicast STUB routing
*=========================*
- Prevents periodic flood and prune behaviour over low-bandwidth links.
- Typical example how this could be asked:
 > Prevent any PIM neighbor relationship on segment A, but retain multicast connectivity and traffic passing
      to end hosts behind the one multicast router on segment A.
- Conceptually similar to DHCP relay.
- Remote site forwards IGMP join requests to central site.

-----------
 COMMANDS
-----------
 #ip pim neighbor-filter {acl}                 - On the central router, filters all PIM messages based on the ACL.
                                               - Prevents PIM adjacency to form
 #ip igmp helper-address {ip}                  - On the stub router, forwards all IGMP messages to central router.



*-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-*
*=====================*
  Filtering
*=====================*
- Static RP 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] On clients specify 1.1.1.1 to be RP for ACL-44 groups
```

```
  Auto-RP Filtering
*--------------------*
 > Candidate-RPs can limit their RP announcements to ONLY include certain multicast groups.

CONFIG-SET: Auto-RP - Candidate 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 Mapping Agent can filter the c-RP's and their c-RP advertisements, to allow only what is accepted.
      !NOTE: The Mapping Agent "rp-announce-filter" MUST match announcements by the cRP specified with "send-rp-announce".
      !NOTE: This can be seen on the MA by using a "debug ip pim".

CONFIG-SET: Auto-RP - Mapping Agent filtering c-RP's
+---------------------------------------------------
|     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 RP's 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 RP's
|


CONFIG-SET: Two-ways to filter Auto-RP Messages with the multicast boundary command
+-----------------------------------------------------------------------------------
|     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 e0/0
|     ip multicast boundary 1                         - 1> Older command requires the ACL
|     ip multicast boundary filter auto-rp            - 2> Newer command doesn't require ACL
|
```

```
CONFIG-SET: Filter admin multicast groups will still allowing IGMP joins to be received
+-------------------------------------------------------------------------------
|     access-list 1 permit 239.0.0.0 0.255.255.255
|     !
|     interface e0/0
|      ip multicast boundary 1                        - Filters all admin scope mtraffic beyond e0/0
|


- BSR Filtering
 > Filtering the BSR messages on an interface can be done with the "ip pim bsr-border" command.
 > Allows exchange of PIM message, but not BSR messages.

- PIM Neighbor Filtering
 > Restrict PIM neighbor establishment on an interface, while still allowing multicast clients to join groups
 > Configured with "ip pim neighbor-filter {acl}".

- Client filtering
 > Used to limit join/prune messages destined for a specified rendezvous point (RP) and for a specific list of groups
 > Configured with "ip pim accept-rp {rp-address | auto-rp} [access-list]".

- By default a host can join any multicast group, it wishes to, on a segment running IP multicast routing.
 > To control which groups a host can join, an ACL can be used with the command "ip igmp access-group".

- To limit the number of multicast routes (mroutes) which can be added to a multicast routing table,
      use the "ip multicast route-limit" command in global configuration mode.

- 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}: Deny statements prohibited, only specify permits
                                                - {interval} how often the candidate announcements are sent

#ip pim rp-announce-filter rp-list {rp-acl} group-list {group-acl}
                                                - Enables the mapping agent 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 all Auto-RP traffic
#ip pim bsr-border                               - Allows exchange of PIM message, but not BSR messages
#ip pim neighbor-filter {acl}                    - PIM Neighbor filter
#ip pim accept-rp {rp-address | auto-rp} [acl]  - Limits client join/prune messages
                                                 - Configured on clients to ignore the RP they don't trust
#ip igmp access-group {acl}                      - The multicast groups which hosts can join on an interface
#ip multicast route-limit {amount}               - Limits the number of mroutes that is allowed to be added to the multicast table

#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 on each outgoing interface.
 > If the Packet TTL >= Interface TTL, then the packet is forwarded.
 > If the Packet TTL < Interface TTL, then the packet is dropped.
 > TTL Scoping is limited, because the configured interface TTL applies to all multicast packets.

- Administrative Scoping
 > You can apply a filter on the interface only allowing certain group addresses in the range.

-----------
 COMMANDS
-----------
#access-list 1 deny   239.0.0.0 0.255.255.255   - Pay attention to the mask.
#access-list 1 permit any
#interface e0/0
 #ip igmp access-group 1                         - This will deny any multicast traffic in the administrative scope.

 #ip multicast ttl-threshold {value}             - Means that any packets TTL lower than the specified threshold
                                                     are not forwarded. Usually used to provide a border to keep
                                                     internal multicast traffic from drifting out of the intranet.
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=================================*
  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
+---------------------------------------------------------------------------------------------------------
|R4>                                                        >>> 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      - Convert the broadcast traffic to multicast, using 239.1.1.1
|
|
|R5>                                                        >>> MULTICAST to BROADCAST client config
|     access-list 123 permit udp any any eq 3001            - Matches the broadcast application traffic
|     !
|     ip forward-protocol udp 3001                          - Change udp 3001 traffic to be processed-switched traffic,
|     !                                                         which is required to the helper-map command
|     interface s0/0                                        - Interface receiving the multicast traffic
|      ip multicast helper-map 239.1.1.1 10.1.1.255 123     - Convert traffic back to broadcast, destination is 10.1.1.255
|     !
|     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 needs to know which
      multicast group address and port are 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
 > 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 Equal Cost Multipaths (ECMP).
 > 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                       - Displays the (S,G) channel subscription through IGMPv3
#show ip mroute                                 - Displays 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
       > IP
        > 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 rendezvous points (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 multiprotocol BGP (MPBGP) for interdomain operation.
- It is recommended that MSDP is run on the RP's 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                          - Displays the contents of Source-Active messages

#ip msdp peer {IP} remote-as {ASN}              - Configures MSDP peer in different AS
#ip msdp peer {IP} connect-source {Int}         - Configures MSDP peer within the same AS
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*====================================*
 PGM (Pragmatic General Multicast)
*====================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > IP
        > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
         > Configuring PGM Host and Router Assist

- PGM is a reliable multicast transport protocol for applications which 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 transport session identifier (TSI) which identifies a particular PGM session.
- The PGM Router Assist feature saves bandwidth substantially.

-----------
 COMMANDS
-----------
# sh ip pgm router                                - Displays information about PGM traffic statistics and TSI state.
# clear ip pgm router                             - Clears the PGM traffic statistics.

#int 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
       > IP
        > 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 near 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 10.1.1.2
|     access-list 2 permit 10.1.4.2
|     !
|     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:test1/10.1.2.2 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:
>            10.1.1.2
>          Test receiver:
>            10.1.4.2
>
>     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 10.1.2.2         10.1.4.2        1            (4%)      29
>     *Feb  4 18:29:54 10.1.2.2         10.1.4.2        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

#int fa0/0
 #ip mrm {test-sender|test-receiver}            - Configures a sender or receiver
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================================================*
   MVR (Multicast VLAN Registration)
*=======================================================*
- DOC-CD LOCATION
       > Switches - LAN
        > Cisco Catalyst 3560 Series Switches
         > Configuration Guides
          > Configuring IGMP Snooping and MVR
           > Configuring MVR
```

- It 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 receive VLANs.
- Clients are in the receive VLANs and the multicast server is in the source VLAN.

- 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                                      - Displays 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
   > IP
    > Cisco IOS IP Multicast Configuration Guide, Release 12.4T
     > Configuring DVMRP Interoperability

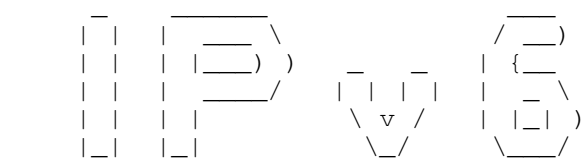
- Cisco IOS does not support a full implementation of DVMRP; however, it does support connectivity to a DVMRP network.
- Cisco routers know enough about DVMRP to successfully forward multicast packets to and receive packets from a DVMRP neighbor.
- It is also possible to propagate DVMRP routes into and 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 which is similar to RIPv2. It sends route updates every 60 seconds and considers 32 hops as 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.
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
  NTP via Multicast
*=====================*

CONFIG-SET: Multicast NTP
+-------------------------------------------------------
|R2   interface Gi0/1
|      ntp multicast 225.0.0.1 ttl 16 version 3 - Setup the NTP multicast server
|      ntp master 2
|     !
|R1   interface fa0/0
|      ntp multicast client 225.0.0.1           - Setup the NTP Client
|      ntp multicast version 3
|     !
>     #show ntp associations [detail]           - Verify the NTP source, detail and other info



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
  Troubleshooting Multicast               >>>   {} curl-brackets indicates replaceble values      <<<
*================================*

*********************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*********************************************************************************************************************************


- For general troubleshooting, consider the following:
 > Have you tried emulating an IGMP join on a client routers interface?                       # sh ip igmp groups
 > Have you tried emulating the multicast traffic from the source?                            # ping {m-ip}
 > Does all transit routers have multicast-routing enabled?                                   # sh ip multicast | i Routing

 > Are you sending and receiving multicast traffic on a 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?                                                             # sh ip igmp interface
 > Does all 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 neighbors configured for the same PIM version?                               # sh ip pim int
 > Are the expected multicast route 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                                          # mstat {src-ip} {m-ip}
 > 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,
      >> Were 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 from all the multicast routers.
 > 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 RPs 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.
 > Is the expected incoming interface and outgoing interfaces listed?            # sh ip mroute
 > Is the incoming multicast interface the next-hop back to the 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 debugs, 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.
 > Uses a static multicast route to force multicast to RPF out 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.


- ERRORS :
 > %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 RP's                                  # sh run | i pim.*accept
```



-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

```
      + Bestpath Selection
- Tunneling & Transitioning Techniques
      + IPv6IP
      + GRE
      + Automatic 6to4
      + ISATAP
      + NAT-PT
- IPv6 Multicast
      + Ethernet Mapping
      + MLD
      + PIM
      + BSR
- Access-List Filtering
- Static IPv6 DNS Entries
- Troubleshooting IPv6




*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
     Overview
*======================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > 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 MTU)
 > 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.

- Aggregatable Global Unicast Addresses
 > 2000 - 3FFF    : Format Prefix
 > Structure consists of
      >> 48-bit Global Prefix assigned to regional registries.
      >> 16-bit subnet ID or Site-Level Aggregator (SLA).
      >> 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 follow:
      >> 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 2 fields: flags and scope. These 2 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
 > An anycast address is 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
 > Is 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 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
 > Lets 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> Thus will give a IPv4 address of 191.168.99.1


- With IPv6 multiple ipv6 addresses can be configured per interface. No primary and secondary like 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                                   - Same as "sh ip arp"
# sh ipv6 route                                      - Same as "sh ip route"
# sh ipv6 int brief                                  - Same as "sh ip int brief"
# sh ipv6 traffic                                    - Displays statistics about IPv6 traffic

# debug ipv6 packets                                 - Displays detailed messages for IPv6 packets
# debug ipv6 nd                                      - Displays 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}        - Telneting to a link-local host required to be sourced

#ipv6 unicast-routing                                - Enables IPv6
#ipv6 cef                                           - Enables CEF for IPv6 (Default = disabled)
#interface fa0/0
 #mac-address 1034.5678.9012                        -(o) Used the specified MAC appose to the built in address (BIA)
 #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 address 2001:oDB8:c058:6301::/128 anycast    - Configures the anycast address

#interface fa0/3
 #ipv6 address autoconfig                           - Address is then based on stateless auto-config
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*========================*
    ICMP v6
*========================*
- ICMPv6 neighbor and router discovery
 > Replaces IPv4 ARP

- Router discovery is the functionality in IPv6, where 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
       keep track of them.

- NS - Neighbor Solicitation
      >> Ask for information about neighbor.
- NA - Neighbor Advertisement
      >> Advertise yourself to other neighbor.
- RS - Router Solicitation
      >> Ask for information about local routers.
- RA - Router Advertisement
      >> Advertise yourself as an active router.

- The sending of RA (Router Advertisement) 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
 > Is 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                               - New command of above



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   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
 > Requires static resolution on multipoint interfaces.
 > This is required for global unicast addresses and link-local addresses else 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                       - Map 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 resolves the final destinations.
 > Point-to-point links requires no next-hop resolution.

- Static to next-hop
 > With ICMP ND (Neighbor Discovery) there is no proxy ability to learn the remote neighbor (like InARP discovery)
 > When a static route is pointed out an interface, it should be pointed to the next-hop instead of the interface.

!!! Dynamic information recurses to remote link-local address, not the global unicast address!!!!

-----------
 COMMANDS
-----------
# sh ipv6 static [detail]                             - Displays information about the IPv6 static routes

#ipv6 route {address} {prefix} {int} {NH address}     - Creates a static IPv6 route



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
     RIPng
*======================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > Cisco IOS IPv6 Configuration Guide, Release 12.4T
          > Implementing RIP for IPv6


- Changes from RIPv2
 > Tags are just locally significant arbitrary numbers/words.
 > Uses UDP port 521.
 > Multicasts 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                                    - Will show if RIP is enabled, and on which interfaces
# sh ipv6 rip                                          - Displays RIP protocol statistics and counters
# sh ipv6 route rip                                    - Displays 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


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
     OSPFv3
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > Cisco IOS IPv6 Configuration Guide, Release 12.4T
          > Implementing OSPF for IPv6

- Still uses protocol number 89.
- Requires separate routing processes for multiple instances.
- 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 now possible per interface.
- Operation is still very similar to OSPF v2.
- One requirement is that the router-id should still be a valid IPv4 address.
 > Should either have a 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 Security Policy Index (SPI).
 > The SPI functions similarly to key numbers in a key chain, but is communicated via AH and must match between
       both ends of the adjacency.
 > The SPI number is arbitrary, but must be between 256 and 4,294,967,295 (32-bit).
```

```
- Two new LSA (Link State Advertisements) were added:
 > Link LSA
      >> Advertises the link-local address to all routers that are attached to the link.
      >> Advertises IPv6 prefixes on the link to the routers that are attached to the link.
      >> Advertises options.

 > Intra-Area LSA
      >> Associates a list of IPv6 prefixes with a transit network by referencing a network LSA.
      >> 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                               - Displays the security associations
# show crypto ipsec policy                           - Displays 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



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    IPv6 - EIGRP
*=====================*
- Uses protocol number 88.
- Uses multicast address- FF02::A.
- A ping the multicast address could be used to verify IPv6 neighbors.
- To configure EIGRP for IPv6, you must enable IPv6 on the interface and unshut the EIGRP routing process.
- EIGRP for IPv6 has a shutdown feature, (Yip Cisco calls it a feature).
      The routing process should be in "no shut" mode in order to start running.

- The router-id used for 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 source address.
```

```
-----------
 COMMANDS
-----------
# sh ipv6 eigrp {asn} neighbors                        - Displays the neighbors discovered, holdtime, uptime, SRTT, RTO, etc
# sh ipv6 eigrp {asn} topology                         - Displays entries in the EIGRP topology table
# sh ipv6 route eigrp                                  - Displays 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} {percentage}      - Configures the bandwidth percent EIGRP may use on a interface. (Def = 75)
 #ipv6 summary-address eigrp 1 2001:0DB8:0:1::/64      - Examples of a 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



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
    IPv6 - MPBGP
*=====================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > IP
         > Cisco IOS IPv6 Configuration Guide, Release 12.4T
          > Implementing Multiprotocol BGP for IPv6


- Only one BGP process is allowed, IPv6 config is done using the address-family configuration.
- IPv6 in BGP is implemented via Multi-Protocol BGP (MPBGP).

- Two new BGP attributes were defined in MPBGP:
 > MP_REACH_NLRI
 > MP_UNREACH_NLRI

- The first two values in these two attributes contain the Address Family Identifier (AFI) and
      the Subsequent Address Family Identifier (SAFI).
- The first value (AFI) identifies the network layer protocol.
- The second value (SAFI) identifies additional information about the type of NLRI carried.

- 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-labelled VPN.

- If BGP is carrying IPv6 traffic, AFI equals 2, SAFI equals 1 for unicast, or SAFI equals 2 for multicast.

- When BGP peers set up the session between them, they send an OPEN message containing optional parameters.
- One optional parameter is capabilities. The possible capabilities are multiprotocol extensions, route refresh,
      outbound route filtering (ORF), and so on.
- When the BGP peers exchange the multiprotocol extension capability, they exchange AFI and SAFI numbers and thus identify what
      the other BGP speaker is capable of.

- Normal BGP rules still apply for MPBGP
 > MPBGP requires a 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.
 > Same best-path selection process using the BGP attributes.

- A IPv6 neighbor must be activated under the address-family. By default it is disabled (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 100
 #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
        > IP
         > 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 hasno 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, since its 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
|      !
|      ipv6 route 2002::/16 tunnel 0                   - Ensures any other traffic to 2002::/16 is directed to tunnel
|                                                             interface 0 for automatic tunneling.
|

- ISATAP
 > Usage: Point-to-multipoint tunnels which 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, since it's 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 a 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

 > An translation mechanism, allowing IPv6-only devices to communicate directly with IPv4-only devices and vice versa by 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:
     >> Port Address Translation (PAT), aka overload, allows a single IPv4 address to be used among 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:
     >> Send traffic from their 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}                             - Displays the interfaces state, counters, etc
# sh ipv6 tunnel                                        - Displays IPv6 tunnel information
# sh ipv6 nat statistics                                - Displays NAT-PT statistics
# sh ipv6 nat translations [verbose]                    - Displays active NAT-PT translations
# clear ipv6 nat translation *                          - Clears dynamic NAT-PT translations
# debug ipv6 nat [detail]                               - Displays debugging messages for NAT-PT translation

#interface tunnel 0                                     - Configure 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 2 fields: Flags and Scope.
 > These 2 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 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 Source-Specific Multicast (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 operates the same as v4-PIM with only a few differences:
 > IPv6-PIM has two modes of operation: sparse mode (SM) and source-specific multicast (SSM).
 > 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 rendezvous point (RP) 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 will automatically associate the IPv6 address of a RP with a multicast group. It will adapt to changes
      in RP mappings in case of failure.


-----------
 COMMANDS
-----------
# sh ipv6 mroute                                    - Displays the contents of the IPv6 multicast routing table
# sh ipv6 mroute active                             - Displays 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                                - Displays the multicast groups directly connected and learned via MLD
# sh ipv6 mld groups summary                        - Displays the number of (*, G) and (S, G) membership reports
# sh ipv6 mld interface                             - Displays multicast-related information about an interface
# sh ipv6 mld traffic                               - Displays the MLD traffic counters


# sh ipv6 pim traffic                               - Displays the PIM traffic counters
# sh ipv6 pim interface                             - Displays information about interfaces configured for PIM
# sh ipv6 pim neighbor [detail]                     - Displays the PIM neighbors discovered
# sh ipv6 pim group-map                             - Displays an IPv6 multicast group mapping table
# sh ipv6 pim bsr {election | rp-cache | c-rp}      - Displays information related to PIM BSR protocol processing
# sh ipv6 mld ssm-map                               - Displays SSM mapping information

# 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} {source}     - Configures MLD reporting for a specified group and source
 #ipv6 mld static-group {group} {incl|excl} {source}   - 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 set the RP address
                                                       - Optionally 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-list {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 fa3/0
 #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
        > IP
         > 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.
```

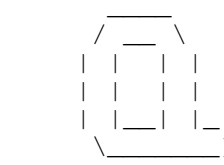
```
CONFIG-SET: IPv6 ACL's
+-------------------------------------------------------
|      ipv6 access-list example1
|          permit tcp any any                           - Allows any TCP traffic regardless of whether or not an AH is present
|      !
|      ipv6 access-list example2                        - 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             >>>   {} curl-brackets indicates replaceble values      <<<
*============================*
******************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
******************************************************************************************************************************

- 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 ACL's blocking protocol number 41?                                                          # sh ipv6 interface | i line|list
 > IPv6 IPv6IP and GRE-IPv4 tunnels
      >> Are  the tunnel source and destinations IPv4 addresses?                                       # sh run int tunnel {t-int}
      >> Is the tunnel address a IPv6 address?                                                         # sh run int tunnel {t-int}

- When troubleshooting IGPs for IPv6, apply the same troubleshooting as with IPv4!
 > For RIPng
      >> Is 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 RIPng database?                              # sh ipv6 rip database
      >> Are the RIPng routes appearing in the table?                                                  # sh ipv6 route rip
      >> Are individual routes of a summary not suppressed?                                            # sh ipv6 rip | i {prefixes}
      >> If only a default route was to be sent out 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 hello?                                                    # debug ipv6 ospf hello
      >> Are the timers matching?                                                                      # sh ipv6 ospf int {int} | i Dead
      >> Are the MTU values matching?                                                                  # 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



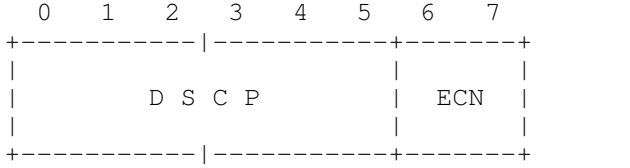
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

```
- Compression
      + TCP Header
      + Predictor
      + RTP Header-compression
      + Frame-Relay RTP compression



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
     QOS Overview
*=========================*
- The TX-Ring/Hardware queue is always FIFO. It can be seen with the "sh controllers" command.
- QOS affects how traffic is processed in the output queue/software queue before the hardware queue.
- Queueing is always applied outbound to the interface.
- Shaping is always applied outbound to the interface.
- Policing can be applied inbound or outbound to the interface.

- The default input hold-queue limit is 75 packets. 10 packets for async interfaces.
- The default output hold-queue limit is 40 packets. 10 packets for async interfaces.
- A length of 1000 will normally resolve problems caused by input queue drops of TCP ACKs, but will introduce bigger delay.

-----------
 COMMANDS
-----------
#sh controllers Se0/0 | 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 queue length for an interface
 #load-interval seconds                                 - Sets the length of time used for load calculations
 #hold-queue {length} {in|out}                          - This command 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 = (3bits IP PREC + 5bits 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



DIFFSERV FIELD VALUES = (6bits DSCP + 2bits 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	0×0
CS1	001 000	8	001 000 00	32	0×20
AF11	001 010	10	001 010 00	40	0×28
AF12	001 100	12	001 100 00	48	0×30
AF13	001 110	14	001 110 00	56	0×38
CS2	010 000	16	010 000 00	64	0×40
AF21	010 010	18	010 010 00	72	0×48
AF22	010 100	20	010 100 00	80	0×50
AF23	010 110	22	010 110 00	88	0×58
CS3	011 000	24	011 000 00	96	0×60
AF31	011 010	26	011 010 00	104	0×68
AF32	011 100	28	011 100 00	112	0×70
AF33	011 110	30	011 110 00	120	0×78
CS4	100 000	32	100 000 00	128	0×80
AF41	100 010	34	100 010 00	136	0×88
AF42	100 100	36	100 100 00	144	0×90
AF43	100 110	38	100 110 00	152	0×98
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 Class-Selector 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 Class-Selector 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)
 > The AF behaviour 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 4 separate classes in the AF PHB (Per Hop Behaviour).
 > 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 six-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 more 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 higher drop value.
 > But if AF33 and AF21 was 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, will be 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 more preferred)

-=-
=====================
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 are 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, a 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
 > Are 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-map can be referenced, which will be 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
 > Fr-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
 > Is arbitrary number locally significant to the router.
 > Is used when traffic passing through the router must be tagged/classified without changing anything in the packet header.

- Nested MQC policies
 > Are used to configure QOS inside other QOS policy-maps.
 > Are often used on sub-interfaces, as sub-interfaces do not have 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                          - This will be the 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 a 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 that transaction
      identifier, message type, or other similar data.
- NBAR natively supports many predefined application/protocols, which can be seen with "match protocol ?"
- A PDLM (Packet Description Language Modules) 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 transits 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 3 match criteria's:
      > 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 goto 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 entry 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
                                                           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
                                                           or 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
*===========================*
- First-In, First-Out (FIFO)
 > Is the default queueing mechanism on ethernet and serial links above 2mb.

- Modified Deficit Round Robin (MDRR)
 > Priority queueing mechanism for 12xx routers.

*------------------------------*
  Weighted Fair Queue (WFQ)
*------------------------------*
- 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 are 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 Custom Queue (CQ)
*-----------------------------*
- 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 behaviour 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 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 Priority Queue (PQ)
*------------------------------*
- 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 gets serviced from high-to-low.
 > PQ is prone to starvation.
 > The queueing strategy will be 'priority-list' as listed with "show interface" command.
 > Similar 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 - MQC Bandwidth
*------------------------------*
- 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 64 classes/queues.
 > Drop policy is tail drop or WRED, and it is configurable per class.
 > Scheduling within a single class:
      >> FIFO on 63 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 class-default needs “fair-queue” configured if “bandwidth” was 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 what is 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 ACL's 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.
 > NOTE: "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) - MQC Priority
*-------------------------------------*
- 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, 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, else 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' as with "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

- Attempt to avoids congestion before it occurs by selectively dropping traffic, ie random-detect.
- Weights are based on IP precedence/DSCP.
- WRED is typically used to avoid TCP global synchronization and generally not to successful when majority of flows are UDP.
- Minimum threshold is when WRED becomes active and starts randomly dropping packets.
- 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 packets are dropped.
- MPD (Mark Probability Denominator).
 > Is used to determine how aggressively packets will be dropped.
 > The lower the number the more aggressively dropped.
 > When max-threshold reached, 1/MPD will be dropped!!!!

*--------------------------------------------------*
  Legacy WRED (Weighted Random Early Detection)
*--------------------------------------------------*
-----------
 COMMANDS
-----------
# sh queueing int {int}                         - Shows the input and output queue size, and default values

#interface s0/0
 #random-detect [dscp-based | prec-based]       - Enabled RED on an interface, by default will be classified by IP precedence
 #random-detect prec {value} {min-t} {max-t} {mpd} - Changes the default values of WRED
 #random-detect dscp {value} {min-t} {max-t} {mpd} - Changes the default values of WRED, (min=10, max=40, mpd=10)
```

```
*-----------------------------------------------*
   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 delay 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 the amount 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.
 > 2 Types of Shaping
      >> Generic Traffic Shaping (GTS)
      >> Frame-Relay Traffic Shaping (FRTS)
 > 2 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. (Can't be changed)

- Shaping CIR
 > Dictates the average output rate one aims to average per second on the circuit/interface.

- Tc (Time Interval)
 > It is 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.
 > The get the TC value correct for the formulas below, always use TC/1000.
 > The maximum value of Tc is 125ms (1/8th of a second) and the minimum value is 10ms (1/100th of a second).
 > 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 SVC's 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 affect on the delay/time interval.

- Bc (Committed Burst)
 > Is the number of committed bits allowed to be sent per interval (Tc) to conform with target-rate (CIR) per second.
 > If Bc worth of bits are sent every interval in that second, 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, as a result the amount of data to send per interval:
  >> Bigger Bc - more delay but more data per Tc.
  >> Smalled Bc - less delay but less data per Tc. (Smaller Bc 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 available credits.
 > If all the Bc per interval was not used, then at a later time the router can send Be 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 its only from the previous interval.
 > Be defaults to zero bits.

- Formulas (Tc/1000):
 > CIR = Bc / Tc
 > Tc = Bc / CIR
 > Bc = CIR x Tc
 > Be = (CAR - CIR) x Tc

  Legacy GTS

- Is used to control the maximum output target rate on an interface.

```
-----------
 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)
*-------------------------------------------*
- CIR
 > Dictates the average output rate one aims to average per second on the circuit/interface.

- 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)
 > Sent towards the destination, to indicate congestion was experienced on the way, which will get reflected back to the source
      as a BECN.

- BECN (Backward Explicit Congestion Notification)
 > Is sent back to the source sending the traffic as a indication to slow down the sending-rate, as there is congestion in the
      direction the traffic is 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 25% each Tc until
      BECN's 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 that the provisioned rate, or
 > To throttle down higher speed site so that it does not overrun a lower speed site, typically used in partial mesh topologies.

- Careful once FRTS is enabled on an interface:
 > All DLCI's on that interface (including sub-interfaces) are assigned the default CIR value of 56000 bps.
 > If DLCI's require a different output rate that 56k, the CIR should be adjusted.
```

```
- If FRTS is applied to a physical frame interface the config will apply to all VC 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 (ie 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 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 fecn-adapt                        - Enables shaping reflection of a received FECN as BECN
 #frame-relay fragment {bytes}                  - Specifies the maximum fragment size
 #frame adaptive interface-congestion {queue-depth}
                                                - If the output queue depth exceeds the configured amount, slow down rate

#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)
*-------------------------------------*
- 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.
- CB-shaping defaults to a Bc and Be = target-Rate * Tc(25ms).

- 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 just applied to a frame-interface
|
```

```
*----------------------------------------*
  MQC FRTS (Frame-Relay Traffic Shaping)
*----------------------------------------*
- Once FRTS has been enabled on the interface, all DLCIs on that interface (including sub-interfaces) are assigned
      the default CIR of 56kbps.

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 max threshold of bandwidth.
 > To accomplish this, a system of credits is used.
 > Before a packet can be sent the amount 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, 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 2 rate policer.
- Legacy CAR statement supports the continue feature to have nested rate-limits.
- Similar to traffic shaping, changing the burst size determines how often the rate is enforced over the second.
- NOTE that rate-limit Bc/Be are in BYTES, unlike shaping where Bc/Be are in bits.
- NOTE Excess burst is only used when the configured Be is greater than the configured Bc.
 > Example with a Bc=1000 and Be=1000 there will be no burst.


- The TC is typically 1 second.
- Formula
 > Bc = CIR/8 * Tc


SYNTAX
 => rate-limit {in|output} [access-group] {CIR (bps)} {Bc (bytes)} {Be (bytes)} conform {OPTIONS} exceed {OPTIONS}
 => OPTIONS
      > continue                        Scans other rate limits.
      > drop                            Drops the packet.
      > set-dscp-continue               Sets the DSCP and scans other rate limits.
      > set-dscp-transmit               Sets the DSCP and sends it.
      > set-prec-continue               Rewrites packet precedence, scans other rate limits.
      > set-prec-transmit               Rewrites packet precedence and sends it.
      > set-qos-continue                Sets QOS-group and scans other rate limits.
      > set-qos-transmit                Sets QOS-group and sends it.
      > transmit                        Transmits the packet.
```

```
-----------
 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, holds no meaning here because conforming
                                                      traffic gets marked with DSCP-12 and so does exceeding traffic

 #rate-limit output access-group 123 128000 24000 48000 conform-action continue exceed-action drop
                                                - Example how to limit traffic matching ACL-123 to 128k

 #rate-limit output 192000 36000 72000 conform-action transmit exceed-action drop
                                                - Example of a "line-rate" statement, configuring the TOTAL output 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 it.
 > set-dscp-transmit                     Sets the DSCP and sends it.
 > set-frde-transmit                     Sets the FR DE and sends it.
 > set-mpls-exp-imposition-transmit      Sets the exp-bits at tag imposition and sends it.
 > set-mpls-exp-topmost-transmit         Sets exp-bits on topmost label and sends it.
 > set-prec-transmit                     Rewrites the packet precedence and sends it.
 > set-qos-transmit                      Sets the QOS-group and sends it.
 > transmit                              Transmits the packet.
```

```
-----------
 COMMANDS
-----------
#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 Cisco IOS 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 layer 3 control packets have priority over other packet types that are destined for the control plane.
- The following types of layer 3 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 control plane policing for all CP packets that are received from all line-card
      interfaces on the router.
- Distributed control plane services provide control plane policing 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
      >> Is traffic which is directly destined for one of the routers 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
      >> Is 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
      >> Is 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 class "icmp-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                 - Displays 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

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

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================================*
  RSVP (Resource Reservation Protocol)
*============================================*
- DOC-CD LOCATION
      > 12.4T Configuration Guides
       > Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4T
        > Part 5: Signalling
         > RSVP

- RSVP on it own is just a reservation tool in the control plane, still require external mechanism to enforce the mechanism.
- Allows end user application to make bandwidth reservations inside the network.
- When using "ip rsvp bandwidth" on a 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 {interface-kbps} {single-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 quality of service (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 to be established.
- 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 10 milliseconds (ms) 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
 > Same as above, previous QOS policies has 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 Voice over IP (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]                      - Displays 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 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 a 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 on all traffic incoming 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 send.
 > On ingress queues, SRR performs sharing among the two queues according the weights configured.
 > The weights are relative rather than absolute, ie like percentage based rather than bandwidth.
 > Firstly specify the ratio's 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 configure 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 percent
      then drops traffic with COS 4 and 5 at 60 percent full, and finally drops COS 6 and 7 traffic
      only when the queue is 100 percent full.
 > WTD is configurable separately for all six queues in the 3560 (two ingress, four egress)

*----------------------------*
  Traffic Policing
*----------------------------*
 > Can be applied 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
|     !                                              >> Step1: 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                    >> Step2: References ACL's to match required traffic
|     class-map SMTP
|      match access-group 125
|     !
|     policy-map QOS                             >> Step3: 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                                    >> Step4: Enables SW-QOS
|     int fa0/5
|      service-policy input QOS                  - Step5: Applies the policy to the interface

----------
 COMMANDS
----------
# sh mls qos                                    - Displays global QOS configuration information
# sh mls qos maps dscp-mutation [name]          - Displays the current DSCP mapping entries.
# sh mls qos maps dscp-cos                      - Displays the DSCP-to-COS map
# sh mls qos interface [buffers|queueing]       - Displays the QOS information at the port level
# sh mls qos input-queue                        - Displays the settings for the ingress queues
# sh mls qos aggregate-policer                  - Displays 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
 > Is 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 to be reduced 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 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 was 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
*=======================*
*********************************************************************************************************************************
***  To efficiently troubleshoot, an in-depth understanding of a protocol, its phases/state, and its operation is required.   ***
***  The points listed here is merely a guideline to offer a structured troubleshooting approach to a knowledged individual.  ***
*********************************************************************************************************************************

- 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 ACL's?                                      # sh class-map {name}
 > If ACL's are used for classification,
      >> Does the ACL exist? (Matching a non-existing ACL = MATCH all traffic)               # sh ip acce {acl}
      >> Are the ACL matching the correct IP's 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 was the amounts specified in kbps?                                            # sh run policy-map | i bandwidth
 > With CB-shaping was the correct rate, Bc and Be values specified?                        # sh run policy-map | i shape
 > With FRTS was 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
      >> Was the Bc and Be specified in bytes?                                              # sh run policy-map | i police
      >> Was the Be configured 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
      >> It is required unless 'passive' is used.

- How to troubleshoot whether or not packets 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.
 > Lets 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 2 zeros.
      >> Convert the remaining 6 bits to decimal. 010010 -> 18.
      >> Thus 48-HEX provides a DSCP decimal value of 18 or AF21.
      >> Is the value you expected to see that was marked at least one router earlier than the show command?
```

```
 > 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
*=====================*
CB-SHAPING:
#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, ie Bc value in bytes
- Adapt Active                 = Shows adaptive shaping has been enabled. If a BECN is received, the flow will be throttled back
```

-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-

```
- NTP
      + Master
      + Server
      + Peer
      + Authentication
      + Time-zones
- Banners & Menus
      + Configuring Banners using tokens
- HTTP Server
- TFTP Server
- FTP Server
- CDP
- Crash Dump
- Warm Reload
- System Resources
      + Memory Threshold Notification
      + CPU Threshold Notification
- KRON Command Schedule
- EEM (Embedded Resource Manager)


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*======================*
     SNMP
*======================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Network Management
         > Network Management Config Guide
          > SNMP Support

- SNMP uses ports UDP-161 and UDP-162 for traps.
- SNMP is used to report conditions of managed devices to management stations (NMS).

- 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 send 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 behaviour can be changed with "snmp-server ifindex persist".
 > To see the interface index numbers "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)
      >>> A full MIB object name will be ifEntry.10.5

- Two ways to collect data
 > Polling
      >> A NMS asks managed devices to report on variables.
      >> Uses SNMP community string, which is a password used by the NMS to poll the device.
      >> 2 Types of community strings:
            + Read Only: Information gathering only.
            + Read Write: Gathers information and can set values.
 > Traps
      >> 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
+-----------------------------------------------------
|     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]         - (step1) Enables all/some snmp traps
                                                      - By specifying the type, only specified traps are enabled
#snmp-server host {ip} {community} [notification-type]- (step2) 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
       > 12.4 Mainline 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
 > Alarm
      >> Conditions that trigger an event
            + CPU exceeding 90%.
            + Free memory droping below 20MB.
 > Event
      >> Messages to send to NMS/syslog.

- DELTA sampling
 > Is a method to sample the selected variable and calculate the value to be compared against the thresholds.
 > Is the difference between MIB values at time index A compared to MIB value at time index B.
      >> Amount of packets sent our ethernet0/0 each minute.
      >> CRC errors on the interface.
 > 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 value that increase and decrease.
```

```
CONFIG-SET: SNMP RMON
+-----------------------------------------------------
|     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 would be triggered if thresholds are exceeded,
|                                                               and generate the specified events

-----------
 COMMANDS
-----------
# sh rmon events                                   - Displays the RMON event table
# sh rmon alarms                                   - Displays the RMON alarm table

#snmp-server host 1.1.1.1 CISCO                    - Enables traps to SNMP server
                                                   - Specifies SNMPv1/v2c community string or SNMPv3 user name

#rmon event {no} {log|desc|trap|owner} {community}    - [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

*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================*
    Logging - Syslog
*=====================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Network Management
         > Network Management Config Guide
          > 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 level. If logging level 3 is enabled, level 2,1,0 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 (see Table 1) is stored in the history table even if syslog
      traps are not enabled.
```

```
- Config Change Notification (aka Archive)
 > DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > System Management
         > Configuration Fundamentals Configuration
          > 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

#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
       > Network Management
        > Cisco IOS NetFlow Configuration Guide, Release 12.4T
         > Part1: Cisco IOS NetFlow Configuration Guide
```

- Good article here: http://blog.ru.co.za/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
 > Layer 3 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 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                                 - Displays the interfaces which netflow is enabled on
# sh ip cache flow                                     - Displays a summary of the netflow statistics, IP's, ports, protocols, etc
# sh ip cache verbose flow                             - Displays 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 destination {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|destination|source|protocol|tos}   - Specifies a match criterion
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=============================*
   Remote-Access via Telnet
*=============================*
- DOC-CD LOCATION
      > 12.4 Mainline Configuration Guides
        > System Management
         > Configuration Fundamentals Configuration
          > Operating Characteristics for Terminals

- DOC-CD LOCATION (Login-Block)
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Security Configuration Guide
          > Login-Block


- A router can be configured to display a message when a console or terminal is not in use, this is called a vacant message.
- Saving local settings between sessions is done with the "private" command.
- Suppressing onscreen messages during telnet connections is done with "ip telnet quiet"

- IOS Login Enhancements (aka login-block)
 > The login block capability, when enabled, applies to both telnet and SSH connections, and more recently to HTTP connections.
 > Attempted DOS attack: a malicious user may attempt to interfere by flooding a device with connection requests
 > Dictionary attack: is 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 seconds after 15 attempts within period
|     !                                                          of 100 seconds
|     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
|


----------
 COMMANDS
----------
#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 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
#term [no] monitor                                 - Enables/disables the display of log messages to telnet session
```

```
#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
 #transport output {none | telnet | ssh}              - Prevents or limit outbound telnet
 #transport preferred none                            - Will prevent the router resolving mistyped commands via DNS
 #busy-message hostname [d message d]                 - Customizes the info displayed during telnet connection attempts
 #vacant-message [d message d]                        - Configures the system to display an idle terminal message
 #refuse-message [d message d]                        - 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

 #login [local] [tacacs]                              - Specifies the password source, terminal password used by default
                                                      [local] use the local username database

                                                      >> 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
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=========================*
   Remote-access via SSH
*=========================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing User Services
          > 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


#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
  > 12.4 Mainline Configuration Guides
   > Security and VPN
    > Cisco IOS Security Configuration Guide: Securing User Services
     > Secure Shell (SSH)
      > Secure Copy (SCP)

- The Secure Copy (SCP) feature provides a secure and authenticated method for copying router configuration or router image files.
- SCP relies on Secure Shell (SSH), an application and a protocol that provide 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 {network|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



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================*
    NTP
*=====================*
- DOC-CD LOCATION
 > 12.4 Mainline Configuration Guides
  > Network Management
   > Cisco IOS Network Management Configuration Guide, Release 12.4
         > 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 source, not the other way around.

- 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.
      >> Is a peer-to-peer relationship. The host will also retain time-related information.
      >> Should be used when there are a number of mutually redundant servers that are interconnected.
      >> Use the "ntp peer" command to specify the NTP peer to consider synchronizing with.

- Broadcast mode
 > Is 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 configured with "ntp broadcast version".
 > The NTP broadcast client configured with "ntp broadcast client".
```

- NTP access-list based restriction scheme allows certain access privileges to 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 a NTP server.
- The time can also be configured manually with "clock set" but this will be reset when the system is restarted.
- The use of manual configuration should only be used as a last resort.
- Optionally time-zones could be configured.

CONFIG-SET: NTP - Client Authenticating a Server

```
|Configures the NTP client to authenticate the Servers 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 authenticate                                    - Enables authentication
|     ntp authentication-key 1 md5 CISCO                  - Defines the authentication keys
|     !
|     ntp master 3                                        - Configures the stratum number. Lower is better!
|     ntp access-group serve-only 10                      - Only allow update to clients matching the ACL
|
```

CONFIG-SET: NTP - Broadcast Server and Client setup

```
|R1: Configured as the broadcast server and R2 as a client
|
|     ntp master 5
|     ntp authentication-key 58 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
```

```
----------
 COMMANDS
----------
# sh ntp status                                        - Displays 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-offset} [minute-offset]  - 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 {id}] [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
```

```
*-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====-=====*
*====================*
   Banners & Menus
*====================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > System Management
         > Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4
          > Part 4: Banners and Menus
```

```
- Banners
 > 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
 > Display the currently configured value of the token argument (eg, the router host name, domain name, or IP address)
      >> $(hostname)    - Router host name
      >> $(peer-ip)     - IP address of the peer machine
      >> $(gate-ip)     - IP address of the gateway machine



CONFIG-SET: Configuring a Custom IOS Menu
+-----------------------------------------------------
|     menu TS title @ MY RACK @                          - Specifies 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 3 Go to R3
|     menu TS text s show all sessions                   - 's' = command s
|     menu TS text c<no> clear the sessions              - 'c<no>' = command c11
|     menu TS text e menu-exit
|     menu TS text q Quit terminal server session
|     menu TS prompt [d title d]                         - This is the prompt!
|     !
|     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     - IE if 2 is pressed, then telnet to the hostname R2
|     menu TS command 3 resume R2 /connect telnet R3
|     menu TS command e menu-exit                        - Allows a 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 ts1 command c11 disconnect 11                 - Disconnects session 11
|     menu ts1 command q exit                            - EXITS from terminal-server completely
|


----------
 COMMANDS
----------
#banner [exec | incoming | login | motd |  prompt-timeout]  - Configures the specified banner
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
   HTTP Server
*======================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Network Management
         > Cisco IOS Network Management Configuration Guide, Release 12.4
          > 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 authentication {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
       > 12.4 Mainline Configuration Guides
        > System Management
         > Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4
          > 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 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'.
 > The default boot image name would be:
      >> 2600 : 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 {NAME}] [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
       > 12.4 Mainline Configuration Guides
        > System Management
         > Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4
          > Part7: Configuring Basic File Transfer Services

- FTP can also be used to transfer files between systems on the network.
- FTP is more preferred than TFTP, because of higher file transfer rate.
- For large IOS upgrades use FTP, eg 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
*======================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Network Management
         > Cisco IOS Network Management Configuration Guide, Release 12.4
          > Cisco Discovery Protocol (CDP)

- For Frame-Relay encapsulated interface, CDP is not enabled by default on the physical interface, only on the sub-interface.

----------
 COMMANDS
----------
# sh cdp                                             - Displays all the CDP protocol info
# sh cdp interface                                   - Displays CDP enabled interfaces
# sh cdp entry {device}                              - Displays 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 (def=60)
#cdp holdtime 90                                     - Changes the CDP hold timer interval (def=180)
#no cdp run                                          - Disables CDP on a supported device (def=enabled)
#int fa0/0
 #no cdp enable                                      - Disables CDP on a supported interface (def=enabled)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=====================*
   Crash Dump
*=====================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Network Management
         > Cisco IOS Network Management Configuration Guide, Release 12.4
          > Troubleshooting and Fault Management

- A crash dump is used for analysis when a router 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
        > System Management
         > 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 reuses the saved data together with IOS code already residing in RAM to restart IOS.
- Of course, the IOS code (depending on 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. 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
```

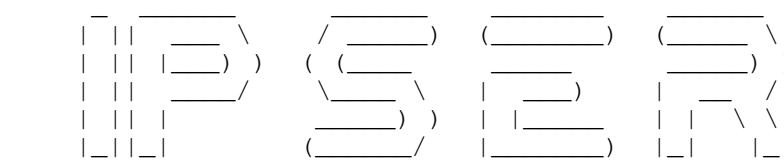
```
----------
 COMMANDS
----------
# sh warm-reboot                                         - Displays statistics for attempted warm reboots
# sh region | i saved                                    - Shows the amount of memory used and address blocks

# 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
       > 12.4 Mainline Configuration Guides
        > Network Management
         > Cisco IOS Network Management Configuration Guide, Release 12.4
          > System Monitoring and Logging

- CPU Threshold Notifications
 > Notifies users when a predefined threshold of CPU usage is crossed by generating a SNMP trap message for the
      top users of the CPU.
 > Two types of CPU utilization threshold are supported:
      >> Rising Threshold
            - If 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 the SNMP section)

- Memory Threshold Notifications and Reservation
 > 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 example 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 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 reservation: 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, log a syslog message when its free processor memory falls below 1 Mbyte, and reserve 512 Kbytes of
|      memory for the notification process itself.
|
| Answer:
|     process cpu threshold type total rising 50 interval 5       - If the CPU threshold rises above 50%, for more than
|     !                                                                 5 seconds, then trigger a notification
|     memory free low-watermark processor 1000                    - Specifies a threshold of 1000 KB of free processor
|     !
|     memory reserve critical 512                                 - Reserves 512 KB 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                                    - Displays the status and schedule of occurrences.

#policy-list {P_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 {P_NAME}                                - Attach the policy-list to execute
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==================================*
   EEM (Embedded Event Manager)
*==================================*
- DOC-CD LOCATION
       > 12.4 T Configuration Guides
        > Network Management
         > Cisco IOS Network Management Configuration Guide, Release 12.4T
          > Embedded Event Manager (EEM)


- EEM is 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 Tool Command Language (TCL).


- 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):
 > CLI Event Detector                        - Detects various CLI commands types based on regular expressions.
 > IP SLA Event Detector                     - Detects when an IP SLA reaction is triggered.
 > NetFlow Event Detector                    - Detects when 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 Routing Information Base (RIB).
 > 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
 > Is the CLI-based corrective actions that are taken when event detectors report events. Enables a powerful on-device
      event management mechanism.
 > Event actions availability depends on 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.



- 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 the loopback from being shutdown
+-------------------------------------------------------------
|     event manager applet Lo0                                - Creates and registers the applet with EEM
|      event syslog occurs 2 pattern "Loopback0.*down"        - Configures 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 "blog@ru.co.za" from "test@lab.com" subject "lo0"
|                 body "someone is playing as per $_cli_result"
|
```

```
----------
 COMMANDS
----------
# sh event manager environment                          - Displays the EEM environment variables set
# sh event manager detector <name> [detail]             - Displays the variables detector
# sh event manager policy registered                    - Displays the EEM policies that are currently registered
# sh event manager history events                       - Displays detailed information about each EEM events
# sh event manager history traps                        - Displays 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/clearl0.tcl                           - References a TCL script

# event man run <applet>                                - Manually executes a none-event applet. (Requires event set to none)


#event manager environment {variable-name 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
```



-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-

```
            o Disabling on router
            o Disabling on switch
      + IP Source Guard
      + Multiple Default Gateway
- DNS
      + Static Entries
      + DNS Client Config
      + DNS Proxy
      + Simple DNS Server Config
      + Authoritive DNS Server Config
            o Mx Record
            o A Record, etc.
- IP Accounting
      + Output Packets
            o Output Filter
      + Access-List Violations
      + Precedence
- RITE (Router IP Traffic Export)
- First Hop Redundancy
      + HSRP
      + VRRP
      + GLBP
- IRDP
- IP SLA/RTR and Track
- Aliases
      + IP Alias
      + Command Alias
- TCP/UDP Small Services
- Web Caching Content Protocol (WCCP)
      + In/Outbound
      + Excluding Traffic from Redirection
      + Using Access-lists for a Service Group
      + Setting a Password for a router and cache engine
      + Enabling on 3550
- DRP Server Agent
- Mobile IP
      + Local Area Mobility
- IP Event Dampening
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*==============================*
  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 process context activity.
- "scheduler allocate 3000 1000" is typically decent setting.

----------
 COMMANDS
----------
#scheduler allocate {network-interrupts} {processes}  - Allows some control between interrupt processing vs. process mode


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================================*
  TCP Performance and Other Features
*=====================================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Application Services Configuration Guide
          > 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 that are called Long Fat Networks (LFNs). The TCP Window Scaling enhancement provides that support.
      > The window scaling extension in 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).
 > They are sent once every minute on otherwise idle connections. If 5 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". The default is 30 seconds.


----------
 COMMANDS
----------
# show tcp brief [all]                                   - Displays 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)



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*============================*
   Nagle
*==============================*
- When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke
       typed, which can use up bandwidth and contribute to congestion on larger networks.

- 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.


----------
 COMMANDS
----------
#service nagle                                           - Enables the nagle slow packet avoidance algorithm
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*===================================*
  MTU (Maximum Transmission Unit)
*===================================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Application Services Configuration Guide
          > Configuring TCP


- MTU (Maximum Transmission Unit)
 > Is 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 maximum segment size (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 negotiate to make sure no fragmentation would be needed once 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
 > By default with IPv4, TCP Path MTU Discovery is disabled. If enabled, the minutes default is 10.
 > By default with IPv6, 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 limited, known as the Maximum Segment Size (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                      - To see 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}                                      - To change (MSS) for TCP connections originating or terminating on a router
                                                        - Disable 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, with a TTL=0.
      >> Each router in the path decrements the IP TTL by 1.
      >> If a router in the path decrements the received packet's TTL to 0, it will discard the packet and
            generate a ICMP time-exceeded, 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 previous TTL increased 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 a unused local udp port, it will respond with a ICMP
            port-unreachable message. Once the source received the ICMP port-unreachable, it knows the destination
            was reached.
      >> If ICMP was used, (same process as before), but the destination will reply with a ICMP echo-reply.

- Three different implementation of traceroute:
 > ICMP
      >> Used native 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
```

```
- Allowing ping and traceroute traffic in ACL's
 > 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 a 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, due to 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 protocol data unit (PDU) was received.
      C     - A congestion experienced packet was received.
      I     - User interrupted test.
      M     - A destination unreachable error protocol data unit (PDU) was received.
      and   - Packet lifetime exceeded.

- ICMP Rate-limit
 > A built-in feature, to protect a router against spoofed ICMP denial-of-service attacks, by rate-limiting the amount of ICMP
      responses out 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 msec  7 msec  5 msec
      2 192.168.10.1  16 msec *  16 msec


----------
 COMMANDS
----------
# traceroute [prot] [dst-ip] [source] [numeric]       - To analyse the path to a destination
# ping [prot] [ip] [df|size|source|timeout|repeat]    - To diagnose basic network connectivity

# show ip icmp rate-limit                             - Displays 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' milliseconds
                                                      - (default = One reply every 500 ms)
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*========================================*
     NAT (Network Address Translation)
*========================================*
- Cisco NAT Whitepaper : http://tinyurl.com/nat-whitepaper
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Addressing Services Configuration Guide
          > Part6: NAT


- 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:
 > Inside Local (IL)    - The local IP address of the private host on a network. Typically from private address space (RFC-1918).
 > Inside Global (IG)   - The public, registered IP address that the outside network sees as the IP address of your local host.
 > Outside Local (OL)   - The destination IP address, which the local host sees as the IP address of the remote host.
 > Outside Global (OG)  - The public, legal, registered IP address of the remote host (eg the IP address of the remote
      web server that a PC is connecting to).

- Be careful when enabling nat, to not nat everything, else locally generated traffic like routing protocol traffic will
      also get natted. If the routing protocol traffic comes from an unknown source, routing protocols will break.

- Static NAT
 > A particular Inside Local address always maps to the same Inside Global (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 natting whole IP's, NAT could also be used to NAT individual TCP/UDP ports, aka 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 IP's.
 > The words "inside source" emphasize that the inside source address is what's getting changed.
 > NAT-POOL Implementation
      >> Many Inside Local address are mapped to a POOL of Inside Global (public) IP addresses.
      >> IP's 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 Inside Local address are mapped to ONE Inside Global (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 Outside Global will be changed before entering the network to the Outside Local.
 > 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 seconds (24 hours)
      icmp-timeout   : 60 seconds   (1 minute)
      udp-timeout   : 300 seconds   (5 minutes)
      tcp-timeout   : 86,400 seconds (24 hours)
      dns-timeout   : 60 seconds   (1 minute)
      syn-timeout   : 60 seconds   (1 minute)
      finrst-timeout   : 60 seconds   (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 natted to the new server, without 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            - That 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 Outside Source Global to Outside Local 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
```

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

- DHCP stands for Don't Hit Computer People!
- DOC-CD LOCATION
  > 12.4 Mainline Configuration Guides
   > IP
    > Cisco IOS IP Addressing Configuration Guide
     > 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 "ip helper-address" command.

- Gothas to look out for!
 > Be sure to excluded IP's 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.
 > When configuring DHCP and earlier in the switching section you configured DHCP snooping, 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:
   Step 1 Configure the router to exclude its own IP address, and other necessary IP's from the DHCP pool.
   Step 2 Configure the DHCP pool, gateway, and name-servers, and other options.
   Step 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 his 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 - To allow 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.


- Interface Broadcast
 > To change an interface broadcast IP addresses from 255.255.255.255 to a subnet broadcast IP address. 185.1.1.255 use the
      "ip broadcast-address 185.1.1.255"


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                  - Displays packet level detail
# debug ip dhcp server events                   - Displays DHCP events and negotiations

#frame-relay interface-dlci 555 protocol ip {IP}- Allows preconfiguring 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 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}                       - Relays bootp request to a DHCP server

#interface fa0/3
 #ip dhcp relay information trusted            - Enables forward DHCP requests that contain option 82 info

#interface fa0/4
 #ip broadcast-address {ip}                    - Changes the interface broadcast IP
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*======================*
     DNS
*======================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Addressing Configuration Guide
          > Part - 4 DNS


- Static 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
```

```
- Authoritive 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


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
    IP Accounting
*======================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Application Services Configuration Guide
          > Configuring IP Services

- Used for the following:
 > To track how many IP packets are received or sent out of an interface,
 > How many packets violate an access-list policy configured on an interface, OR
 > Track packets with an 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. VERY USEFUL!!!

#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


*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=====================================*
  RITE (Router IP Traffic Export)
*=====================================*
- DOC-CD LOCATION
       > 12.4T Configuration Guides
        > Security and VPN
         > 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 inline, 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 you to export one in every few packets in which you are interested.
      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]         - Displays 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
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=========================*
    First Hop Redundancy
*=========================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Application Services Configuration Guide
          > Configuring First Hop Redundancy


- HSRP (Hot Standby Router Protocol)
 > Cisco proprietary.
 > One ACTIVE router replies to ARP requests sent to 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 seconds.
 > Hold time = 10 seconds.
 > Can use MSEC with HSRP v2.
 > Highest priority preferred, default = 100.
 > Roles can only be changed only if pre-emption is enabled, (disabled by default)
 > Authentication - plain text and MD5.
      >> A plain text password of 'cisco' is 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 virtual IP address.
 > Transport protocol 112, multicast destination address 224.0.0.18.
 > Interface level command "vrrp".
 > VRRP work almost identical to HSRP with one big exception, with VRRP pre-emption is enabled by default.
 > VRRP master = HSRP active.
 > Pre-emption enabled by default.
 > Master 82 advertisement interval = 1 second default.
 > Master down interval = 3.609 seconds 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 pre-emption
 > 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 seconds, then the track kicks in

                                                      >>> HSRP-CONFIG <<<
#interface e0/0
 #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(sec|msec)}{hold-time} - Changes the hello and hold time
 #standby [group] priority {1-255}                    - Changes the priority. Higher preferred, (default = 100)
 #standby [group] preempt                             - Enables pre-emption
 #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 {value}    - If track even is successful, decrement the priority with configured value


                                                      >>> VRRP-CONFIG <<<
#interface e0/1
 #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


                                                      >>> GLBP-CONFIG <<<
#interface fa0/0
 #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 {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 pre-emption
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*========================================*
   IRDP (ICMP Router Discovery Protocol)
*========================================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Application Services Configuration Guide
          > 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
*================================*
- aka RTR (Response Time Reporter)
- aka SAA (Services Assurance Agents)

- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Network Management
         > Cisco IOS IP SLAs Configuration Guide, Release 12.4

- Trackable parameters:
 > Delay (UDP/VoIP)
 > Application response times (HTTP/DHCP/DNS/FTP)
 > Reachability (ICMP echo / UDP Echo / TCP Connect)

- IP SLA can be used for:
 > Statistic reporting
 > HSRP/VRRP/GLBP tracking
```

```
- Enhanced Object Tracking (aka track command) extends basic tracking to
 > Interface line protocol status
 > IP address lost (DHCP/IPCP)
 > Routing reachability
 > Routing metrics

- TRACK DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Application Services Configuration Guide
          > Configuring Enhanced Object Tracking

-----------
 COMMANDS
-----------
# sh ip sla statistics
# sh ip sla monitor operational-state

#ip sla monitor 1                                    - Create a SLA monitor
 #type pathEcho protocol ipIcmpEcho {IP} source {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 threshold    - 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



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*=============================*
    IP and Command Aliases
*==============================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Additional and Legacy Protocols
         > Cisco IOS Terminal Services Configuration Guide, Release 12.4
          > 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 aliases 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




*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*===========================*
   TCP/UDP Small Services
*===========================*
- TCP
 > By default, the TCP servers for Echo, Discard, Chargen, and Daytime services are disabled.
 > When the minor TCP/IP servers are disabled, access to the Echo, Discard, Chargen, and Daytime ports cause the Cisco
      IOS software to send a TCP RESET packet to the sender and discard the original incoming packet.


- UDP
 > By default the UDP servers for Echo, Discard, and Chargen services are disabled.
 > When the servers are disabled, access to Echo, Discard, and Chargen ports causes the Cisco IOS software to send an
      "ICMP port unreachable" message to the sender and discard the original incoming packet.


-----------
 COMMANDS
-----------
#service tcp-small-servers                             - Enables the TCP small servers
#service udp-small-servers                             - Enables the UDP small servers
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*==========================================*
   WCCP (Web Caching Content Protocol)
*==========================================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Application Services Configuration Guide
          > 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.

- 2 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                                      - Will display the current SDM template and stats
# sh ip wccp interface

#sdm prefer extended-match                           - Required on 3350 to support 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
```

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



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
     Mobility
*======================*
- LAM (Local Area Mobility)
 > Offers a simple way for users to roam around the network.
 > When the command "ip mobile arp" is issued on a interface, the LAM process starts listening for ARP requests received on the
      interface that are from hosts which are not in the IP subnet of that interface.
 > When these requests are received, the hosts IP address is then installed in routing table as a mobile host route.
 > ARP requests are sent to the host at a more frequent interval (minutes instead of hours) in order to ensure the host
      is still there.
 > LAM is not a scalable technology.
 > By default any host on the segment can be mobile, alternatively a access-group could limit the mobile IP's


CONFIG-SET: LAM (Local Area Mobility)
+--------------------------------------------------
|     access-list 2 permit 192.1.2.3                   - Creates a ACL for allowed mobile hosts
|     !
|     int fa0/0
|      ip mobile arp access-group 2                    - Enables LAM on the interface and reference ACL-2
|     !
|     router rip
|      redistribute mobile metric 1                    - Advertises the mobile host to the rest of the network
|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
    IP Event Dampening
*=========================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > IP
         > Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4
          > Configuring IP Routing Protocol-Independent Features

- Configures a router to automatically dampen a flapping interface, use the dampening command in 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 seconds.
 > 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                               - Displays a summary of interface dampening
# sh interface dampening                               - Displays 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
```

THIS PAGE WAS LEFT BLANK INTENTIONALLY

-=-=-=-=-=-=-=-=-=-=-=-=-
| INDEX |
-=-=-=-=-=-=-=-=-=-=-=-=-

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=========================*
   Standard Access-Lists
*=========================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4
          > Access Control Lists (ACLs)

- Used to match only on source IP addresses.
- Numbered ACL ranges: 1-99, 1300-1999.



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================================*
  Extended Access-lists
*======================================*
- Numbered ACL ranges: 100-199, 2000-2699
- 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)

- 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


   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 {no} {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 3rd 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 2nd, 3rd & 4th octet 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 2nd, 3rd & 4th octet 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 2nd, 3rd & 4th octet 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


    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 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 is the smallest possible amount, with (2^x) where x = number of one's (1) in the mask.
 4- Convert the binary mask to decimal to get value of the octet in question.


EXAMPLE STEPS: Create a one line ACL to match both of these networks: 168.208.3.0/24 & 168.192.3.0/24
 1- Convert second octet in question to binary:
      192   11000000
      208   11010000

 2- MASK >  00010000 = only the 4th bit differs between the two.

 3- Possibilities, 1bit difference = 2^1 = 2 possibilities. Smallest > 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 /24's 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 /24's 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 your IP range 150.100.32.0/24.
      #access-list 18 permit 150.100.32.128 0.0.0.126
```

```
   ACL Logging
*-----------------*
- 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
- Log-input includes log options, and source L2 MAC address, and 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 5-minute intervals thereafter.

 > Logging rate-limit of syslog messages
      >> Limits the CPU impact of log generation and transmission.
      >> Applies to all syslog messages.
      >> Limit 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 access-list
*----------------------*
 > Traffic filter with "ip access-group"
 > In/Outbound exec access control with "access-class"
 > Route-filter with "distribute-list"


CONFIG-SET: Local policy routing for local 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 logging                                        - Displays the console buffer
# sh ip access-list {name|number}                  - Displays the configured access-list/s
```

```
# terminal monitor                                      - Displays logging output to the current terminal line, ie the VTY screen
# 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 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 like 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, eg the port numbers etc
#access-list 101 deny icmp any any log-input

#ip access-list logging interval {msec}                 - Specifies every 'ms' a log entry is create when a match occurs
#ip access-list log-update threshold {no of hits}       - Specifies how many acl hits before generating the log entry
                                                        - Default: at 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 Access-Lists
*============================*
- Firstly, know that there are 8 IP precedence values 0-7.
- Construct a bit vector consisting of 8 IPP values:  [p7][p6][p5][p4][p3][p2][p1][p0].

      IPPrec      Rate-Limit binary value
       0           00000001
       1           00000010
       2           00000100
       3           00001000
       4           00010000
       5           00100000
       6           01000000
       7           10000000

- Secondly, if required to match different IP precedence values, let’s say 1, 3 and 7, add the binary values together to get
10001010.
- Convert 10001010 to hex
 > Firstly, take each 4 bits and convert to decimal.
 > Then, convert each decimal to hex notation.
 > For example:   1000 = 8, and
                  1010 = 10 = A in hex
                  Thus 10001010 provides a rate-limit mask of 8A.
- Then the easy part:
      #access-list rate-limit 1 mask 8A
      #int Fa0/0
       #rate-limit input access-group rate-limit 1 1000000 35000 35000 conform-action transmit exceed-action drop
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*=======================================*
   Time-based Access-Lists
*=======================================*

CONFIG-SET: Timed-Based ACL example
+------------------------------------------------------
|     time-range OFFICE                                     - Create you range-group
|      periodic weekdays 9:00 to 16:59                      - Specify the allowed times
|     !
|     ip access-list ext TIMEBASED
|      10 permit tcp any any eq www time-range OFFICE       - References the time-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 Access-Lists
*=========================*
- 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)


- AKA "Lock and Key" ACLs
- Application example to use from the inside:
      >> User must first authenticate, before allowed to send traffic to the internet.
      >> Poor man version of configuration proxy.

- Application example to use from the outside:
      >> User must first authenticate, before allowed to access internal web server.

- Typically two ACL formats (There must be at least ONE DYNAMIC PERMIT entry above a deny statement):
      >> With EXPLICIT permit format:
            1. dynamic permit
            2. static deny
            3. explicit permit
      >> With 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 create entry
>


- To authenticate and test Lock-and-Key
 > Telnet to the lock-and-key router.
 > Authenticate with username and password.
 > See below config-set for 3 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,
      eg, "telnet 195.1.15.3 25"

CONFIG-SET: Dynamic ACL - Activation can be achieved 3 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 behaviour, to disconnect a user after
|                                                                     authentication
|3rd:
|     router> access-enable                                     - Once successfully authenticated, issue the command manually
|                                                                     on lock-and key router to allow access
|
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================================*
   Reflexive ACL
*======================================*
- 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)


- Aka 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
+-------------------------------------------------------
|     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 ACE's
#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 {rname} [timeout]        - Defines the reflexive access list using the reflexive permit entry
 #evaluate {rname}                                         - Creates the dynamic reflect ACE's

#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

- CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information.
- CBAC 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.
- CBAC 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.
- CBAC 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.
- CBAC has the ability to detect and prevent certain types of DOS attacks such as SYN-flooding.
- CBAC is 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 be simply 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
+------------------------------------------------------
|     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 local router to ping out fa0/0 and receive replies
|      permit tcp any any eq bgp
|      permit tcp eq bgp any                            - Have to manually allow routing protocol traffic
|      deny ip any any                                  - Non 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 [alert] [audit] [timeout] - Enables CBAC inspection for TCP packets
#ip inspect name {NAME} udp [alert] [audit] [timeout] - 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 12.4 T
   > Security and VPN
    > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4T
     > Zone-Based Policy Firewall

- The Zone-Based Policy Firewall utilizes CBAC technology, but provides more functionality.
- Typically meant for deployment in branch offices.

- Features:
 > Stateful firewall, Layer 3 through layer 7 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.
 > Support 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
 > Allows 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 behaviour for traffic originated by the router.
 > Take care when doing 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.
 > 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, else all 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. NO remarking option available.
 > There is no optimal value for the burst parameter.
 > A smaller burst, causes less traffic to be sent 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, which control the behaviour 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
  > 12.4 Mainline Configuration Guides
   > Security and VPN
    > Cisco IOS Security Configuration Guide: Securing the Data Plane, Release 12.4
     > Configuring Port to Application Mapping

 > Aka PAM (Port-Application-Mapping)
 > Network applications that use nonstandard 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, which is based off MQC).
- CPL Configuration Steps:
     1- Define zones   - Decide on the interface groupings, eg 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.
     >> Eg voice uses two channels, one for voice and one for signalling.

- Typical performance counters

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                      - Layer 3/4 top-level inspect policy
|      class type inspect HTTP-TRAFFIC                       - Calls the class-map
|       inspect                                              - Define the action
|       police 512000 burst 16000                            - Defines the aggregate police rate
|     !
|     zone security OUT                                      - Creates and label 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      - Displays 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 Layer 4 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 Layer 3 and Layer 4 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, 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, 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 for which the signature was seen for a specified amount of time.
```

```
- Individual signatures can be disabled in 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, which 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 SME's (Signature Micro Engines) to load the SDF and scan signatures.
- Signatures contained within the SDF are handled by a variety of SME's.
- 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 a 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 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

- 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
      IP RTP      -     16384 > 32767

- 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
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*===============================*
   Security Compliance RFC's
*===============================*
RFC 1918
---------
      10.0.0.0/8
      172.16.0.0/12
      192.168.0.0/16

RFC 3330 (more for the SP track)
---------
      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
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Security Configuration Guide: Securing the Data Plane
          > Configuring TCP Intercept

- A SYN flood DOS attack: 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 expired 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 attack, by allowing a router to intercept the initial SYN, and respond with a SYN/ACK. If the ACK was received, the session is forwarded onto the server, else a RST will be generated.

- Used to prevent TCP-SYN DOS attacks.
- Attacked would sent only SYN packets, but never completes the connection.
- 2 modes:
  >> Watch - This mode just monitors the tcp setup, and if half open sessions, will send the SYN/ACK to the receiver.
  >> Intercept - This mode actually proxies the tcp setup and intercept the TCP sessions.

- Optionally, an ACL can be used to restrict which hosts should be watched.

```
-----------
 COMMANDS
-----------
# sh tcp intercept statistics                       - Displays TCP intercept statistics
# sh tcp intercept connections                      - Displays 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
  > 12.4 Mainline Configuration Guides
   > Security and VPN
    > Cisco IOS Security Configuration Guide: Securing User Services
     > IP Source Tracker

- The IP Source Tracker feature allows information to be gathered about the traffic which is 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                        - Displays 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
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing User Services
          > User Security Configuration
           > IP Traffic Export

- Without the ability to export IP traffic, the Intrusion Detection System (IDS) probe must be inline 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 direct all exported traffic to a VLAN that is dedicated for network monitoring.
- By alowing users to choose the optimal location for their IDS probe reduces processing burdens.


-----------
 COMMANDS
-----------
# sh ip traffic-export {interface} {profile}          - Displays 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
#int fa2/1
 #ip traffic-export apply {name}                      - Enables IP traffic export on an ingress interface



*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=*
*========================*
   Disabling Services
*========================*
- Source Routing
 > Allows the source to determine the route the packet will take through the network to reach the destination
 > Enabled by default.
 > 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
 > Can be a security risk, but can also be used for troubleshooting, using the telnet, ping, or trace on CISCO IOS.
 > Disabled with 'no ip source-route'
```

```
 > Example of Source-Route Trace:
      R4#traceroute
      Protocol [ip]:
      Target IP address: 222.22.2.1
      Source address:
      Probe count [3]: 1
      Minimum Time to Live [1]:
      Maximum Time to Live [30]: 10
      Loose, Strict, Record, Timestamp, Verbose[none]: Loose
      Source route: 192.1.0.1 192.1.0.2 192.1.203.3 192.1.35.5 192.10.1.254


- Proxy ARP
 > Enables a router to answer an ARP request if destination IP address is not on the local segment and the router
      has a route for that destination in the routing table.
 > Enabled by default.
 > Disabled with 'no ip 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.
 > Enabled by default.
 > The Complication by disabling Proxy-ARP comes in especially with default routing.
 > When disabled, for each destination the router will try to find the layer3-to-layer2 mapping.

- BOOTP and DHCP
 > BOOTP was developed long before DHCP.
 > BOOTP is disabled with 'no ip bootp'
 > Even when BOOTP is disabled, the router will still listen on UDP-67 if DHCP if enabled.
 > DHCP is disabled with 'no service dhcp'.

- 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.
 > Often used to hide network devices.
 > Enabled by default.

-----------
 COMMANDS
-----------
#no ip source-route                                  - Disables source-routing options
#no ip bootp                                        - Disables (BOOTP) bootstrap server
#no service dhcp                                    - Disables the DHCP service
#interface fa0/0
 #no ip proxy-arp                                   - Disables proxy ARP
 #no cdp enable                                     - Disables CDP for the interface
 #no ip unreachables                                - Prevent the interface from generating unreachables
```

```
*-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-*
*======================*
    URPF
*======================*
- DOC-CD LOCATION
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Security Configuration Guide: Securing the Data Plane
          > 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 to 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.
- Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.
- Unicast RPF 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 Unicast RPF does not find a reverse path for the packet, the packet is dropped or forwarded, depending on whether
      an ACL is specified
- With Unicast RPF, all equal-cost "best" return paths are considered valid. This means that Unicast RPF 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.


- 2 MODES:
 > Strict Unicast RPF mode
      A strict mode check is successful when Unicast RFP finds a match in the FIB for the packet source address and the
      ingress interface through which the packet is received matches one of the Unicast RPF interfaces in the FIB match.
      If this check fails, the packet is discarded. This type of Unicast RPF check can be used where packet flows are expected
      to be symmetrical.

> Loose Unicast RPF 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
      is 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                 - Create the ACL-100 to log denied traffic
|     access-list log-update threshold 10                 - Set ACLs to log every 10th entry
|     !
|     interface Serial 0/0
|      ip verify unicast source reachable-via rx 100       - Enable 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
  > 12.4 Mainline Configuration Guides
   > Security and VPN
    > Cisco IOS Security Configuration Guide: Securing User Services
     > Configuring Security with Passwords, Privilege Levels, and Login Usernames

- Before securing a device, it should understand 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 provide limited access to exec commands. (exec commands being the show and clear commands)
   >> Secure this mode by setting terminal line passwords, ie vty, console, and aux.
   >> Default prompt for this mode is : 'Router>'

 > Privileged EXEC Mode
   >> Also know 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 privilege levels and assigning command and users account to the different levels.

```
- The privilege command is used to move commands from one privilege level to another, in order to create the additional levels
      of administration of a networking device, which 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 user accounts to only see privilege level 2 when logged
|     !
|     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      - Specify 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 l337             - 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
       > 12.4 Mainline Configuration Guides
        > Security and VPN
         > Cisco IOS Security Configuration Guide: Securing User Services
          > Authentication, Authorization, Accounting
```

- Full AAA knowledge out the scope of the R&S lab exam. (Only need to know the IOS config side, there is no AAA servers)
- 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.

- 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 to authorization.
 > radius - RADIUS server is queried to 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.
```