# CCIE Lab Notes

## CCIE Lab Notes

- Create a Layer2 diagram.
  - If not restricted, create all required VLANs on all switches.
  - Document the VLANs in the lab by using VTP transparent (copy from running config to text file)
- Watch out for logging discriminator added to logging.
  - If you enable logging console, remember to restore the way it was.
  - Clear history and logs.
- Be careful to match output precisely (traceroute ip unreachables)
- Remember Split-Horizon settings when configuring EIGRP.
- EIGRP does not need IP routing enabled to form neighbors.
- To use ? in command line (passwords) enter CTRL+V first.
- OSPF E1 routes are preferred over E2, even though they might have higher cost.
- Remember Storm-Control can block multicast traffic between IGP neighbors.
- External major network is another word for classful network (10.0.0.0/8, 172.16.0.0/20, 192.168.0.0/16).
- Summarize with minimum required routing information is another word for most specific summary, not a default route.

```
show run | i prefix|access|kron|event|map|filter|list|nat|policy|police|control

access-list 100 permit icmp any any
debug ip packet 100

ping broadcast address
show ip cef
show adjacency
ping IGP multicast addresses
```

**Virtual devices used in the lab**
Cisco ISR 2900 Series routers running IOS version 15.3T
Catalyst 3560X Series switches running IOS version 15.0SE

# AAA

## Authentication, Authorization and Accounting (AAA)

- Authentication - Who is the user?
- Authorization - What is the user allowed to do?

- Accounting - What did the user do?

In AAA a router/switch is a:
- Network Access Device (NAD)
- Network Access Server (NAS)

The NAD challenges the user for credentials, then passes the credentials along to the AAA server

Verify users through:
- Locally configured usernames
- Users present on radius server
- Users present on tacacs+ server

All of these options can be used at the same time and are consulted in order of configuration

## AAA Authentication

Use the new method list aaa by configuring **aaa new-model**, this does the following for authentication:
- Enable the usage aaa commands
- Change the vty line authentication methods to **login authentication default** (aaa)
  - Also disables the **login** and the **login local** commands
  - This default authentication method list uses local usernames and passwords
- You can create custom method lists using the **aaa authentication login** *list_name* command
  - Manually apply the method lists to the specific line with the **login authentication** *list_name* line command
- Local usernames are not automatically placed in privilege level 15 (even when configured on the username)
- In order to automatically put users in exec mode, configure **aaa authorization …** or hardcode the default line privilege

```
aaa new-model
```

```
aaa authentication login default | list_name local | local-case | line | enable | group radius | group tacacs+ | none
```

**local-case** - Consult the local password database, and enforce case-sensitive username rules
**local** - Consult the local password database
**line** - Use the line password for authentication (no username required)
**enable** - Use the enable password/secret for line authentication
**group radius** - Consult configured radius servers (in configuration order)
**group tacacs+** - Consult configured tacacs+ servers (in configuration order)

**none** - Do not require any form of authentication

- Authentication methods are consulted from left to right
  - Configured tacacs+/radius servers are consulted in the order that they were configured
  - See tacacs+/radius AAA section for more detailed configs
- Place local login method before group authentication so that specific usernames are authenticated first
  - If you don't do this, all requests will be forwarded to the aaa server first (even if the local usernames exist on the device)

```
aaa authentication login default local-case group radius
aaa authentication login CUSTOM_LIST local group tacacs+
```

Do not require any login on VTY lines with custom list

```
aaa authentication login NO_LOGIN none

line vty 0 4
 login authentication NO_LOGIN
 privilege level 15
```

Consult case sensitive username for authentication

```
aaa authentication login default local-case
username Admin privilege 15 password cisco
```

- Even though user Admin has privilege level 15, the user will not be placed in exec mode
  - Configure **aaa authorization ...** or hardcode the default line privilege level

## AAA Authorization

- Use the new method list aaa by configuring **aaa new-model**, this does the following for authorization:
  - Automatically configures the **authorization exec default** on the vty lines
- Using aaa new model, all connecting users will receive privilege mode 1, you can influence/override this behavior with:
  - Set the default line privilege to 15
  - Require users to enter privilege exec mode with the enable password/secret
  - Configure aaa authorization

```
aaa authorization exec | commands | config-commands default | list_name
```

**exec** - Decides whether the user can enter privilege exec mode

**commands** - The server authorizes all commands at the specified privilege level (TACACS+ + local only)
**config-commands** - Authorize commands entered in configuration mode (configure terminal)

- This is enabled by default by configuring **aaa authorization commands …**
- However, you might want the server to only authorize privilege exec commands and not configuration mode commands
    - In this case use **no aaa authorization config-commands**

The most important part of authorization is the ability to enter privilege exec mode

- Unless you want to authorize specific commands (for specific privilege levels) you only need **aaa authorization exec**
    - This command is needed if you want users to immediately enter privilege exec mode without an enable password

```
aaa authorization exec default | list_name local | group radius | group tacacs+ | if-authenticated
```

**local** - Use the local database
**group radius** - Consult configured radius servers (in configuration order)
**group tacacs+** - Consult configured tacacs+ servers (in configuration order)
**if-authenticated** - Authorization if the user has been authenticated

- Confusing command, more used as a fallback method in case the first authentication method fails (server times out)

```
aaa authorization console
```

- The console is treated differently by default, and requires additional commands in order to automatically logout users
    - The **aaa authorization console** command applies AAA rules to the console as well

TACACS+ for authentication and authorization

```
aaa new-model
aaa authentication login default group tacacs+ local

aaa authorization exec default group tacacs+ if-authenticated
aaa authorization commands 15 default group tacacs+
aaa authorization config-commands
```

- Fall back to local user database if TACACS+ server is unreachable during authentication
- Authorize users automatically if TACACS+ server is unreachable during authorization

One method list for console and one for vty lines

```
aaa new-model

aaa authentication login CONSOLE local
aaa authorization exec CONSOLE local
aaa authorization console

aaa authentication login VTY group radius local
aaa authorization exec VTY group radius if-authenticated

line con 0
 login authentication CONSOLE
 authorization exec CONSOLE

line vty 0 4
 login authentication VTY
 authorization exec VTY
```

- Fall back to local user database if RADIUS server is unreachable during authentication
- Authorize users automatically if RADIUS server is unreachable during authorization
- Authenticate and authorize console sessions using local database

## AAA Accounting

- Send log messages to specified server in order to verify admin activity or read system events
- The accounting of specific commands can only be done by TACACS+ servers
- RADIUS servers can provide accounting methods for other services like exec and system

```
aaa accounting exec | commands | system default | list_name start-stop | stop-only | none
group radius | group tacacs+
```

**exec** - Logs when an exec sessions has started and who started it
**commands** - Logs commands that are entered (TACACS+ only)

- Specify the privilege level commands (0-15) that will be logged to the server with the **aaa accounting commands** *0-15* **...** command

**system** - Logs switch events such as a reload

**start-stop** - Events are recorded when they start and stop
**stop-only** - Events are recorded only when they stop
**none** - No events are recorded

Log to RADIUS server

```
aaa accounting exec default start-stop group radius
aaa accounting system default start-stop group radius
```

Log to TACACS+ server

```
aaa accounting commands 15 default start-stop group tacacs+
aaa accounting exec default start-stop group tacacs+
aaa accounting system default start-stop group tacacs+
```

## RADIUS vs TACACS+

| **Characteristics** | **RADIUS** | **TACACS+** |
|---|---|---|
| Protocol / ports | New UDP 1812 (authentication)<br>New UDP 1813 (accounting)<br>Old UDP 1645 (authentication)<br>Old UDP 1646 (accounting) | TCP 49 |
| Modularity | Combines authentication and authorization | Separates authentication & authorization |
| Encryption | Only the password | Entire packet |
| Primary use | Network Access | Administration (accounting / permissions) |
| Standardized | Yes | No (cisco proprietary) |
| Resource | Light usage | Heavy usage |
| Accounting | Basic | Robust |

- See RADIUS and TACACS+ sections for more information and configuration

## AAA Auto-Command Account

- Automatically logout particular users (PPP usernames for example) and prevent them from managing the router
    - The **autocommand** function will only work if authorization is configured using AAA

```
username R1 autocommand logout
```

# Local Users

## Local Usernames and Passwords

The privilege level dictates the permissions a user has (see AAA Privilege section)
In IOS there are 3 default privilege levels:

- Privilege 0
- Privilege 1 - user exec
- Privilege 15 - privilege exec

```
username user privilege 0-15 password string
username user privilege 0-15 secret string
```

If you don't specify a privilege level, the user will have a default privilege (1)

- Instead the user can 'ascend' to the next privilege level by using the **enable** command

```
enable password level 0-15 string
enable secret level 0-15 string
```

- If you don't specify a level, the default will be 15

```
line vty vty_range
 password string
 privilege 0-15
 login
 login local
 no login
```

**password** - Sets the default password (no username) for a line

- This option is disabled when you configure aaa new-model which will require a username + password
- When using aaa new-model, you can re-enable this feature using the **aaa authentication login .... line** command

**privilege** - Sets the default privilege level for a line

- All connections to this line will receive this default privilege level (using just the password above)
- When using local usernames and **login local**, per-user privilege levels override vty default privilege levels

**login** - Enables a line for logging in (with the password above, not a username)
**login local** - Use local usernames and passwords instead of the line password
**no login** - Disable the verification of logins (anyone will be able to connect without supplying credentials)

Do not ask for credentials, anyone that connects will enter privilege exec mode

```
line vty 0 4
```

```
no login
privilege level 15
```

Ask for password only and place anyone that connects in privilege exec mode

```
line vty 0 4
 password cisco
 login
 privilege level 15
```

Ask for local username credentials

```
username cisco password cisco
enable password cisco
line vty 0 4
 login local
```

- User cisco that connects will be placed in level 1
- After connecting, user cisco will need to enter privilege mode (15) with the enable password

Ask for local username credentials

```
aaa new-model
username cisco privilege 15 password cisco
```

- Local authentication is enabled automatically on the vty line (login authentication default)
- User cisco that connects will be placed in privilege mode (15) without needing an enable password

# Login / Passwords

## Modify Login Parameters

```
login on-failure log
login on-success log
login delay 3
login block-for 10 attempts 2 within 15
login quiet-mode access-class QUIET

ip access-list standard QUIET
 permit host 10.0.0.10
```

- Log both successful and failed login attempts
- Customize delay between successive login attempts
- Do not allow login attempts for 10 seconds if two tries fail within 15 seconds
- During login block allow hosts in **QUIET** access-list

Before two failed attempts in 15 seconds

```
show login
     A login delay of 3 seconds is applied.
     Quiet-Mode access list QUIET is applied.
     All successful login is logged.
     All failed login is logged.

     Router enabled to watch for login Attacks.
     If more than 2 login failures occur in 15 seconds or less,
     logins will be disabled for 10 seconds.

     Router presently in Normal-Mode.
     Current Watch Window
         Time remaining: 2 seconds.
         Login failures for current window: 0.
     Total login failures: 0.
```

After two failed attempts in 15 seconds

```
show login
     Router presently in Quiet-Mode.
     Will remain in Quiet-Mode for 7 seconds.
     Restricted logins filtered by applied ACL QUIET.
```

## Secure Shell (SSH)

- Enable SSH without ip domain-name by using the **label** keyword
- Normally the first generated RSA key is linked to SSH, override this with the **keypair-name** command

```
crypto key generate rsa modulus 1024 label CR1
ip ssh version 2
ip ssh rsa keypair-name CR1
username cisco privilege 15 password cisco

line vty 0 4
 transport input ssh
 login local
```

```
show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
```

```
show crypto key mypubkey rsa
% Key pair was generated at: 19:32:17 UTC Jan 21 2017
Key name: CR1
Key type: RSA KEYS
 Storage Device: not specified
 Usage: General Purpose Key
 Key is not exportable.
```

Connect via SSH

```
ssh -v 2 -l admin 10.0.0.1
```

## Password Encryption

- Type 0 = plain text
- Type 4 = SHA-256
- Type 5 = MD5
- Type 7 = Vigenère cipher

- The **service password-encryption** command, encrypts all running-config passwords as Type-7
- Using the **enable password cisco** / **username cisco password cisco** command stores the password in plain-text
- Using the **enable secret cisco / username cisco secret cisco** command stores the password as a hash
    - If you see (enable secret 4 xhWjs….) it is stored as a Type-4 SHA-256 hash (newer devices)
    - If you see (enable secret 5 xhWjs….) it is stored as a Type-5 MD5 hash (older devices / less secure)
- You can directly configure the enable password using a MD5/SHA-256 hash with the **enable secret 4 xhWjs….** command

Crack the Vigenère cipher with

```
ntp authentication-key 1 md5 cisco
enable password cisco
service password-encryption
```

```
show running-config | include enable|ntp

enable password 7 00071A150754
ntp authentication-key 1 md5 030752180500 7
```

```
key chain CRACK
 key 1
  key-string 7 00071A150754
 key 2
  key-string 7 030752180500
```

```
show key chain CRACK
Key-chain CRACK:
    key 1 -- text "cisco"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
    key 2 -- text "cisco"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
```

## Telnet

- Default ToS is 192 (C6)
- Change the Telnet ToS with the **ip telnet tos** command

Hide IP / hostname information when establishing Telnet sessions

```
ip telnet hidden addresses
ip telnet hidden hostnames
```

Or

```
service hide-telnet-addresses
```

Show the line connected to when Telnet establishes

```
service linenumber

show users
show line
```

# Privilege

## Privilege Access Control

- The **all** keyword matches the current command and all underlying sub-commands
    - For example give access to all interfaces and all interface sub-configuration options
- A privilege level also has access to all the commands defined in a lower privilege level
- Privilege level 8 has access to privilege 1-7 and 8
- In IOS there are 3 default privilege levels:
    - Privilege 0
    - Privilege 1 - user exec
    - Privilege 15 - privilege exec (default)

Create custom commands for level 10

```
privilege exec level 10 [commands]
privilege configure level 10 [commands]
privilege interface level 10 [commands]
```

# RADIUS

| **Characteristics** | **RADIUS** | **TACACS+** |
|---|---|---|
| Protocol / ports | New UDP 1812 (authentication)<br>New UDP 1813 (accounting)<br>Old UDP 1645 (authentication)<br>Old UDP 1646 (accounting) | TCP 49 |
| Modularity | Combines authentication and authorization | Separates authentication & authorization |
| Encryption | Only the password | Entire packet |
| Primary use | Network Access | Administration (accounting / permissions) |
| Standardized | Yes | No (cisco proprietary) |
| Resource | Light usage | Heavy usage |
| Accounting | Basic | Robust |

## Radius

- RADIUS servers are consulted in the order in which they were configured
    - All configured servers are used, unless you create custom aaa groups (see below)
- By default, the switch/router waits 5 seconds before a message is timed-out (can be customized)
- By default, the switch will retransmit a message 3 times (can be customized)
- You can configure global RADIUS settings or host-specific
    - Host-specific settings override global settings

- All RADIUS servers configured will become public servers by default
    - Configure private servers (for single purpose) using the aaa server groups (see below)

- Default radius ports are 1645 for authentication and 1646 for accounting
- Newer ports are 1812 for authentication and 1813 for accounting

Configuration methods:

- Old - uses the **radius-server host** command
    - Does not require aaa new-model configured
- New - uses the **radius server** command
    - Requires aaa new-model configured
    - Supports IPv6 servers

Old configuration method

radius-server host *ip* | *hostname* **auth-port** *udp_port* **acct-port** *udp_port* **timeout** *seconds*
**retransmit** *tries* **key** *string*
radius-server **timeout** *seconds*
radius-server **retransmit** *tries*
radius-server **key** *string*

**auth-port** - Authentication port (1645 or 1812 by default)
**acct-port** - Accounting port (1646 or 1813 by default)
**retransmit** - The amount of times RADIUS messages are resent to the server (3 by default)
**timeout** - Change the default (5) timeout value

Two RADIUS servers (one custom)

```
radius-server host 10.0.0.102 auth-port 1812 acct-port 1813 timeout 2 retransmit 2 key cisco2
radius-server host 10.0.0.101
radius-server timeout 5
radius-server retransmit
radius-server key cisco
```

- Server 10.0.0.101 will use timeout 5, retransmit 3 and key cisco on UDP port 1645/1646
- Server 10.0.0.102 will use timeout 2, retransmit 2 and key cisco2 on UDP port 1812/1813

New configuration method

aaa new-model
radius server *server_name*
 address **ipv4 | ipv6** *ip* | *hostname* **auth-port** *udp_port* **acct-port** *udp_port*
 **key** *string*
 **retransmit** *tries*
 **timeout** *seconds*

**auth-port** - Authentication port (1645 or 1812 by default)
**acct-port** - Accounting port (1646 or 1813 by default)
**retransmit** - The amount of times RADIUS messages are resent to the server (3 by default)
**timeout** - Change the default (5) timeout value

- Supports only a single (IPv4 or IPv6) server per defined server
- Uses same default settings as the old configuration method

```
radius-server host 10.0.0.101 key cisco

aaa new-model
radius server RADIUS
 address ipv4 10.0.0.102 auth-port 1812 acct-port 1813
 key cisco
 retransmit 3
 timeout 5
```

```
show radius server-group all
Server group radius
    Sharecount = 1  sg_unconfigured = FALSE
    Type = standard  Memlocks = 1
    Server(10.0.0.101:1645,1646) Transactions:
    Authen: 0   Author: 0       Acct: 0
    Server_auto_test_enabled: FALSE
     Keywrap enabled: FALSE
    Server(10.0.0.102:1812,1813) Transactions:
    Authen: 0   Author: 0       Acct: 0
    Server_auto_test_enabled: FALSE
     Keywrap enabled: FALSE
```

Authorize and Authenticate on the RADIUS server

```
aaa authentication login default group radius local
aaa authorization exec default group radius  if-authenticated
```

Send accounting information to the RADIUS server (logs)

```
aaa accounting exec default start-stop group radius
aaa accounting system default start-stop group radius
```

## AAA Server Groups

- AAA Grouped servers are used for a single purpose
    - Normally all servers present in the global configuration will be consulted in order of configuration

    - With server groups, you can for example use certain RADIUS servers for PPP authentication only
- Server groups can contain either public or private servers:
    - Private servers are used for a single purpose and do not have to exist in the global configuration
    - Public servers are a combination of one or more servers used for a single purpose (have to exist in global configuration)

```
aaa group server tacacs+ | radius group_name
 server name server_name
 server ip | hostname
 server-private ip | hostname
```

**server** - Links to a public server (in the global config) configured with the **radius-server host** command
**server name** - Links to a public server (in the global config) configured with the **radius server** *server_name* command
**server-private** - Creates a private server (IPv4 or IPv6) that does not exist in the global configuration

Public server group for PPP

```
radius-server host 10.0.0.101 key cisco

radius server RADIUS
 address ipv4 10.0.0.102
 key cisco

aaa group server radius PPP_RADIUS
 server 10.0.0.101
 server name RADIUS

aaa authentication ppp default group PPP_RADIUS
```

- The server 10.0.0.101 links to the RADIUS server created with the old-style command
- The server name RADIUS links to the RADIUS server created with the new-style command

Private server group for dot1x

```
aaa group server radius DOT1X_RADIUS
 server-private 10.0.0.103 key cisco

aaa authentication dot1x default group DOT1X_RADIUS
```

```
show radius server-group DOT1X_RADIUS
Server group DOT1X_RADIUS
    Sharecount = 1  sg_unconfigured = FALSE
    Type = standard  Memlocks = 1
    Server(10.0.0.103:1645,1646) Transactions:
    Authen: 0   Author: 0       Acct: 0
    Server_auto_test_enabled: FALSE
     Keywrap enabled: FALSE
```

## RADIUS Change of Authorization (CoA)

- The CoA feature provides a mechanism to change the attributes of a AAA session after it is authenticated
- The change is initiated on the radius server and 'pushed' to the router
- The default port for packet of disconnect is 1700, ACS uses 3379
- The client is the radius server that sends the CoA request
- Requests allow for:
    - Session identification
    - Host re-authentication
    - Session termination

```
aaa new-model
aaa server radius dynamic-author
 client 10.0.0.10 server-key Pa$$w0rd
 port 3799
 auth-type all

show aaa clients
```

# RBAC

## Role-Based Access Control (Parser Views)

- Requires AAA enabled
    - AAA authorization is recommended
- Requires enable 'root' password configured
- A **superview** combines two or more parser views

```
enable secret cisco
enable view root

username cisco view VIEW privilege 15 password cisco

aaa new-model
aaa authentication login default local
```

```
aaa authorization exec default local

parser view VIEW
 secret cisco
 commands exec include configure terminal
 commands configure ...

show parser view
enable view VIEW

parser view SUPERVIEW superview
 secret cisco
 view VIEW1
 view VIEW2
```

# TACACS+

| **Characteristics** | **RADIUS** | **TACACS+** |
|---|---|---|
| Protocol / ports | New UDP 1812 (authentication)<br>New UDP 1813 (accounting)<br>Old UDP 1645 (authentication)<br>Old UDP 1646 (accounting) | TCP 49 |
| Modularity | Combines authentication and authorization | Separates authentication & authorization |
| Encryption | Only the password | Entire packet |
| Primary use | Network Access | Administration (accounting / permissions) |
| Standardized | Yes | No (cisco proprietary) |
| Resource | Light usage | Heavy usage |
| Accounting | Basic | Robust |

## TACACS+

- TACACS+ servers are consulted in the order in which they were configured
  - All configured servers are used, unless you create custom aaa groups (see below)
- By default, the switch/router waits 5 seconds before a message is timed-out (can be customized)
- You can configure global TACACS+ settings or host-specific

    - Host-specific settings override global settings
- All TACACS+ servers configured will become public servers by default
    - Configure private servers (for single purpose) using the aaa server groups (see below)

Configuration methods:

- Old - uses the **tacacs-server host** command
    - Does not require aaa new-model configured
- New - uses the **tacacs server** command
    - Requires aaa new-model configured
    - Supports IPv6 servers

Old configuration method

```
tacacs-server host ip | hostname single-connection port tcp_port timeout seconds key string
tacacs-server timeout seconds
tacacs-server key string
```

**single-connection** - Reuse the existing TCP session

- This is more efficient because the session doesn't have to be rebuild

**port** - Change the default (49) TCP port
**timeout** - Change the default (5) timeout value

Two TACACS+ servers (one custom)

```
tacacs-server host 10.0.0.102 port 4949 timeout 2 key cisco2
tacacs-server host 10.0.0.101
tacacs-server timeout 5
tacacs-server key cisco
```

- Server 10.0.0.101 will use timeout 5 and key cisco on TCP port 49
- Server 10.0.0.102 will use timeout 2 and key cisco2 on TCP port 4949

New configuration method

```
aaa new-model
tacacs server server_name
 address ipv4 | ipv6 ip | hostname
 key string
 port tcp_port
 single-connection
```

**single-connection** - Reuse the existing TCP session

- This is more efficient because the session doesn't have to be rebuild

**port** - Change the default (49) TCP port
**timeout** - Change the default (5) timeout value

- Supports only a single (IPv4 or IPv6) server per defined server
- Uses same default settings as the old configuration method

```
tacacs-server host 10.0.0.101 single-connection port 49 timeout 5 key cisco

aaa new-model
tacacs server TACACS
 address ipv4 10.0.0.102
 key cisco
 timeout 5
 single-connection
```

```
show tacacs public
Tacacs+ Server -  public  :
             Server address: 10.0.0.101
                Server port: 49

Tacacs+ Server -  public  :
                Server name: TACACS
             Server address: 10.0.0.102
                Server port: 49
```

Authorize commands on the TACACS+ server

```
aaa authentication login default group tacacs+ local

aaa authorization exec default group tacacs+ if-authenticated
aaa authorization commands 15 default group tacacs+
aaa authorization config-commands
```

- Specify the privilege level in order to authorize for a specific level (usually 15)

Send accounting information to the TACACS+ server (logs)

```
aaa accounting commands 15 default start-stop group tacacs+
aaa accounting exec default start-stop group tacacs+
aaa accounting system default start-stop group tacacs+
```

- Specify the privilege level in order to log for a specific level (usually 15)

## AAA Server Groups

- AAA Grouped servers are used for a single purpose
    - Normally all servers present in the global configuration will be consulted in order of configuration

    - With server groups, you can for example use certain TACACS+ servers for PPP authentication only
- Server groups can contain either public or private servers:
    - Private servers are used for a single purpose and do not have to exist in the global configuration
    - Public servers are a combination of one or more servers used for a single purpose (have to exist in global configuration)

```
aaa group server tacacs+ | radius group_name
 server name server_name
 server ip | hostname
 server-private ip | hostname
```

**server** - Links to a public server (in the global config) configured with the **tacacs-server host** command
**server name** - Links to a public server (in the global config) configured with the **tacacs server** *server_name* command
**server-private** - Creates a private server (IPv4 or IPv6) that does not exist in the global configuration

Public server group for PPP

```
tacacs-server host 10.0.0.101 key cisco

tacacs server TACACS
 address ipv4 10.0.0.102
 key cisco

aaa group server tacacs+ PPP_TACACS
 server 10.0.0.101
 server name TACACS

aaa authentication ppp default group PPP_TACACS
```

- The server 10.0.0.101 links to the TACACS+ server created with the old-style command
- The server name TACACS links to the TACACS+ server created with the new-style command

Private server group for dot1x

```
aaa group server tacacs+ DOT1X_TACACS
 server-private 10.0.0.103 key cisco

aaa authentication dot1x default group DOT1X_TACACS
```

```
show tacacs private
Tacacs+ Server -  private  :
             Server address: 10.0.0.103
                Server port: 49
```

# Access-Lists

## Access-Lists (ACL)

- Adding the **log** keyword to an ACL entry will create a log entry and will process switch the packet
- This can lead to high CPU load because process switched packets are handled directly by the CPU and no by CEF or the fast switching buffer
- Do not use the log keyword, unless you specifically want to log denied packets that come in on VTY lines for example

## Compiled Access-Lists

- The turbo ACL feature is designed in order to process ACLs more efficiently
- Applies to all access-lists present on the device

```
access-list compiled
show access-list compiled
```

## VTY Access-Lists

- Outbound ACL on interface only filters transit traffic, not locally generated traffic
- Outbound ACL on VTY lines only applies to connections generated from an existing VTY session
- Inbound ACL on VTY applies to outside connections coming into the router

```
ip access-list standard VTY_IN
 10 permit host 192.168.0.1
 99 deny any log

ip access-list standard VTY_OUT
 10 permit host 192.168.0.2
 99 deny any log

ip access-list logging interval 1
ip access-list log-update threshold 1

line vty 0 4
 access-class VTY_OUT out
 access-class VTY_IN in
```

# Dynamic

## Dynamic Access-Lists (Lock-and-Key)

- Blocks traffic until users telnet into the router and are authenticated
- A single time-based dynamic entry is added to the existing ACL
- Idle timeouts are configured with **autocommand**
- Absolute timeouts are configured in the ACL
- The absolute **timeout** value must be greater than the idle timeout value, if using both
- If using none, the access-list entry will remain indefinitely
- It is also possible to set the autocommand access-enable host timeout directly on the VTY line
- Absolute timers can be extended by 6 minutes using the **access-list dynamic-extended** command (requires re-authentication)

```
username bpin password Pa$$w0rd
username bpin autocommand access-enable host timeout 4

ip access-list extended DYNAMIC
 10 permit ospf any any
 20 permit tcp host 10.0.12.1 host 10.0.12.2 eq telnet
 30 dynamic ICMP timeout 8 permit icmp host 10.0.12.1 any
 99 deny ip any any log-input

access-list dynamic-extended

int fa0/0
 ip access-group DYNAMIC in
```

# IPv6 ACL

## IPv6 Access-Lists

- By default IPv6 access-lists add (hidden) permit statements that are necessary for IPv6 to function (neighbor discovery, etc)
- IPv6 only supports named and extended ACLs
- If using an explicit **deny**, make sure to manually include the ND and RA/RS statements

```
ipv6 access-list ALLOW_TELNET_OSPF
 sequence 10 permit 89 host FE80::2 any
 sequence 20 permit tcp host 12::2 any eq telnet
 sequence 30 permit icmp any any nd-na
 sequence 40 permit icmp any any nd-ns
 sequence 50 permit icmp any any router-advertisement
 sequence 60 permit icmp any any router-solicitation
 sequence 99 deny ipv6 any any
```

```
int fa0/0
 ipv6 traffic-filter ALLOW_TELNET_OSPF in
```

# Reflexive

## Reflexive Access-Lists

- Only allow return traffic if inside source initiated the traffic
- The **reflect** keyword links ACLs together, the evaluate entry is dynamically created based on this reflect entry
- By default the dynamic entry will timeout after 60 seconds

```
ip access-list extended TRAFFIC_FROM_R3
 10 permit ospf any any
 20 permit icmp host 192.168.0.3 any reflect ICMP

ip access-list extended TRAFFIC_TO_R3
 10 evaluate ICMP
 99 deny ip any any log-input

ip access-list logging interval 1
ip access-list log-update threshold 1
ip reflexive-list timeout 60

int fa0/0
 description LINK_TO_R3
 ip access-group TRAFFIC_FROM_R3 in
 ip access-group TRAFFIC_TO_R3 out
```

# Time-Based

## Time-Based Access-Lists

- Specify time ranges that will allow traffic during specific periods
- Based on local system time
- Two types:
    - Periodic - recurring time (daily, weekly, etc)
    - Absolute - specific start and end time

**Periodic ACL**

```
time-range DAILY
 periodic daily 09:00 to 17:00
```

```
ip access-list extended ALLOW_ICMP_TIME
 10 permit icmp host 10.0.0.100 any time-range DAILY
 20 deny ip any any log-input

int gi0/0
 ip access-group ALLOW_ICMP_TIME in
```

**Absolute ACL**

```
time-range ABSOLUTE
 absolute start 09:00 1 Jan 2016 end 17:00 31 Dec 2016

ip access-list extended ALLOW_ICMP_TIME
 10 permit icmp host 10.0.0.100 any time-range ABSOLUTE
 20 deny ip any any log-input

int gi0/0
 ip access-group ALLOW_ICMP_TIME in
```

# BGP

## AS Ranges

| Purpose | AS Range | Total AS |
|---|---|---|
| Reserved | 0<br>65535 | 1<br>1 |
| Public | 1 - 64495 | 64495 |
| Documentation | 64496 - 64511 | 15 |
| Private | 64512 - 65534 | 1022 |

## ASPLAIN and ASDOT Notation

| Method | Size (also known as) | Range |
|---|---|---|
| ASPLAIN (older) | 2-byte (octet) / 16-bit<br>4-byte (octet) / 32-bit | 1 - 65535<br>65536 - 4294967295 |
| ASDOT (newer) | 2-byte (octet) / 16-bit<br>4-byte (octet) / 32-bit | 1 - 65535<br>1.0 - 65535.65535 |

There is also a new Private AS range using the new 32-bit AS numbers

- ASPLAIN - 4200000000 to 4294967294
- ASDOT - 64086.59904 to 65535.65534

- Total Private AS 94967294 (over 94 million!)

## Default Routing

- The nature of a default route received from a BGP peer (or a static default route) will forward everything that is unknown
  - Meaning that an internal subnet of 10.0.0.0/8 will forward other private ranges (192.168.x.x/16 / 172.16-32.x/12) towards the ISP
  - To prevent this behavior a discard route can be created to forward all traffic that is unknown -and destined for a private range to null0
  - ip route 192.168.0.0 255.255.0.0 null0
  - ip route 172.16.0.0 255.255.240.0 null0

## Homing

There are four defined ISP connection methods:

| Terminology | Description |
|---|---|
| Single-homed | 1 ISP connection using a single link (interface) |
| Dual-homed | 1 ISP connection using two or more links (interfaces) |
| Single-multihomed | 2 ISP connections using a single link for each ISP (interface) |
| Dual-multihomed | 2 ISP connections using two or more links for each ISP (interfaces) |

- Remember that multihomed is always a connection to multiple ISPs
- Single and dual refer to the amount of links used

## BGP Best Path Selection

- A peer Autonomous System (AS) is what defines an external or internal route
- A route learned from an external peer will be external, until the point where the route is forwarded (or redistributed) into own AS
- First criteria is that the NEXT_HOP is reachable, meaning that it is present in the routing table
- Which protocol was responsible for generating the route is irrelevant
- Cisco uses N-WLLA-OMNI-ORI (see below)

| Preference | Path Attribute | Hint | Description | Scope | Preference |
|---|---|---|---|---|---|
| 0 | NEXT_HOP | N | Next hop needs to reachable | Local | If no route to NEXT_HOP exists, prefix will not be used |
| 1 | WEIGHT | W | Cisco proprietary weight attribute (not a PA)<br>Only locally significant, any prefix learned from neighbor | Local | Higher is better (default is 32768) |

| | | | | | |
|---|---|---|---|---|---|
| | | | has a weight of 0. Can only influence local decisions | | |
| 2 | LOCAL PREFERENCE | L | Only relevant in own AS, can influence other iBGP neighbors | Non-transitive | Higher is better (default is 100) |
| 3 | LOCALLY INJECTED | L | Routes are either locally injected, or learned from a neighbor (iBGP / eBGP) | Local | Locally injected (network, aggregate) over remotely learned (iBGP or eBGP) |
| 4 | AS_PATH | A | The amount of hops, the length of the AS_PATH | Transitive | Less hops is better |
| 5 | ORIGIN | O | The method with which routes were originally (or altered) advertised<br>0 (i) is internal, 1 (e) is external (absent), 2 (?) is redistributed (incomplete) | Transitive | I over e over ?<br>e is absent by default in IOS, usually the result of an alternation using a route-map |
| 6 | MED | M | Multi-exit Discriminator.<br>Used to influence direct neighbor (and only direct neighbor) or peers in own AS<br>Often used by dual-homed ISPs to create active/passive connection<br>Only used when all other attributes match (preference / origin / etc) | Transitive (only one AS) | Lower is better, default is 0 or 'missing' in IOS (see below) |
| 7 | NEIGHBOR TYPE | N | The neighbor type, eBGP or iBGP neighbors | Local | Prefer eBGP over iBGP neighbor paths |
| 8 | IGP METRIC | I | BGP routes will receive a metric when NEXT_HOP is learned through an IGP (eigrp / ospf / static / etc) protocol | Local | Lower metric is better |
| 9 | OLDEST | O | Oldest eBGP route | Local | Older is better |
| 10 | RID | R | Lowest neighbor router-ID (RID) | Local | Lower is better |
| 11 | IP | I | Lowest neighbor IP address | Local | Lower is better |

Order of preference:

- WEIGHT (highest)

- LOCAL_PREF (highest)
- Locally injected (network, aggregate) over remotely learned
- AS_PATH (shortest)
- ORIGIN (lowest) (0 over 1 over 2) 0 is internal, 1 is external (absent), 2 is redistributed (incomplete)
- MUTLI_EXIT-DISC / MED (lowest)
- eBGP over iBGP learned routes
- Lowest IGP cost/metric to the NEXT_HOP

Best-Path Tie Breakers (No Multipath)

- If both paths are external, prefer the older one
- If both paths are internal, prefer the lowest ROUTER_ID
    - If ORIGINATOR_ID is the same, prefer one with the shorter CLUSTER_LIST
    - Finally, prefer the one with the lowest neighbor's IP-address

Multipath is enabled

- Weight, Local preference, AS_PATH length, Origin, MED, AS PATH content must match + eBGP and iBGP specific additional rules

## BGP Synchronization

- Synchronization is only relevant in iBGP (disabled by default) and applies to the entire process
- Routes will only be passed on if they are 'synchronized', this is a loop prevention mechanism and only really useful when BGP routes are redistributed into an IGP (not recommended)
    - Meaning that all iBGP learned prefixes must have an identical route in the routing table learned from an IGP
    - In other words, the same route / prefix must be in the IGP routing table (RIB) populated by EIGRP/OSPF/RIP before it can be used
- The originator of the route in the routing table has a router-id (RID)
    - This IGP RID has to match the BGP neighbor that the prefix was received from otherwise its not synchronized
    - The BGP RID of the route originator has to match the IGP RID
- To fix synchronization problems change either the originators IGP RID or the neighbors BGP RID to the same value
    - Make sure that the RID of BGP matches the RID of OSPF/RIP/EIGRP on the same router

```
router bgp 1
 synchronization
```

## BGP Attributes

- BGP uses Network Layer Reachability Information (NLRI), which is the route and prefix and certain attributes

| | | |
|---|---|---|
| Well known, Mandatory | Must appear in every UPDATE message<br>Must be supported by all BGP software implementations<br>If missing, a NOTIFICATION message must be sent to the peer | NEXT_HOP<br>AS_PATH<br>ORIGIN |

| | | |
|---|---|---|
| Well-Known, Discretionary | May or may not appear in an UPDATE message<br>Must be supported by any BGP software implementation | LOCAL_PREF<br>AT_AGGREGATE |
| Optional, Transitive | May or may not be supported in all BGP implementations<br>Will be passed on if not recognized by the receiver | AGGREGATOR<br>COMMUNITY |
| Optional, Non-Transitive | May or may not be supported in all BGP implementations<br>Not required to pass on, may be safely ignored | MED<br>ORIGINATOR_ID<br>CLUSTER_LIST |

## BGP Attributes Scope

- Transitive, these attributes are across AS boundaries
- Non-transitive, these attributes are restricted to the same AS
- Local, these attributes are local to the router only (weight)
- Transitive attributes can be altered to be local only, non-transitive attributes cannot be altered
- MED can be transitive but only to one neighbor AS, not more

**NEXT_HOP Attribute**

This is a well-know, mandatory, transitive attribute that must be present in all updates

- Is the peers IP-address if remotely learned
- Is 0.0.0.0 for routes advertised using the **network** or **aggregate** commands
- Is the IP next-hop for redistributed routes
- The next-hop must be reachable, meaning that it must be present in the routing table
- Remains unchanged in the same AS by default, but can (or should) be modified
- Is changed by default when forwarded between different AS. Will become the IP address of the router that passed on the route

**AS_PATH Attribute**

This is a well-know, mandatory, transitive attribute that must be present in all updates

- Ordered List (AS_SEQUENCE)
- Read from right to left
- Can have an unordered component AS_SET
- Used for loop prevention only
- Shorter count is better (hop count)
- Local ASN is added when advertised to an external peer
- Can be modified using route-maps
- AS Path Prepending
- In special cases can be shortened using the **neighbor "neighbor-ip-address" remove-private-as** command
    - This will remove the private AS that is connected to a non-private as from appearing in the AS_PATH
    - Neighboring AS will assume that the route originated from the non-private AS

**ORIGIN Attribute**

This is a well-know, mandatory, transitive attribute that must be present in all updates. Origin is not part of AS_PATH

Three possible values:

- IGP (0, shown in IOS as "i")
- EGP (1, shown in IOS as "e")
- Incomplete (2, shown in IOS as "?")

Set at the injection point:

- Using "network" or "aggregate" will result in IGP origin code
- Using redistribution will result in incomplete origin code

Can be modified using route-maps using the **neighbor** statement or at the insertion point

**WEIGHT Attribute**

This is a proprietary, optional, non-transitive attribute that is Cisco only and is only of local significance
Higher value preferred, default values are:

- 32768 for locally inserted
- 0 for remotely learned

Can be changed using neighbor statement (directly) or using route-maps (both only apply in the inbound direction)

```
ip prefix-list WEIGHT permit 172.16.0.0/24 le 32

route-map SET_WEIGHT permit 10
 match ip address prefix-list WEIGHT
 set weight 50
route-map SET_WEIGHT permit 99

router bgp 1
 neighbor 10.0.12.1 remote-as 2
 add ipv4
  neighbor 10.0.12.1 route-map SET_WEIGHT in
```

**Or**

```
router bgp 1
 neighbor 10.0.12.1 remote-as 2
 add ipv4
  neighbor 10.0.12.1 weight 50
```

**LOCAL_PREF Attribute**

This is a well-known, discretionary, non-transitive attribute that must be supported by all vendors, but may not appear in every UPDATE message.

- The LOCAL_PREFERENCE is only of significance to the same AS
- Higher value preferred, default value is 100. Not always visible using show commands
- Takes effect in the entire autonomous system, even in confederations (sub-autonomous systems)
- Most effective way to influence path preference for incoming routes, meaning outgoing traffic paths
- Can be modified using route-maps
- The local preference value is absent from eBGP learned or advertised routes
- The weight is always 0 for all learned routes (unless altered on the local router)
- Updates received from (and sent to) eBGP peers do not include the LOCAL_PREFERENCE PA

- IOS lists a null value for LOCAL_PREFERENCE for eBGP-learned routes by default

```
ip prefix-list LOCAL_PREF permit 172.16.0.0/24 le 32

route-map SET_LOCAL_PREF permit 10
 match ip address prefix-list WEIGHT
 set local-preference 200
route-map SET_LOCAL_PREF permit 99

router bgp 1
 neighbor 10.0.12.1 remote-as 2
 add ipv4
  neighbor 10.0.12.1 route-map SET_LOCAL_PREF in/out
```

**ATOMIC_AGGREGATE Attribute**

- Well-known discretionary attribute
- Must be recognized by all BGP implementations, but does not have to appear in all UPDATES
- This attribute is set when routes are aggregated (summarized) at a BGP speaker and forwarded to another AS
- Alerts BGP peers that information may have been lost in the aggregation and that it might not be the best path to the destination
- For example, if 10.0.1.0/24 and 10.0.2.0/24 are summarized to 10.0.0.0/22 it will include the 10.0.0.0/24 and 10.0.3.0/24
- Aggregated routes will inherit the attributes of the component routes
- When routes are aggregated, the aggregator attaches its RID to the route into the AGGREGATOR_ID attribute, unless the AS_PATH is set using the AS_SET statement
- It is not possible to disable the discard route in BGP

**Communities Attribute**

- This is an optional, transitive attribute that is comparable to route-tags
- Not used for best-path selection
- Large number (32 bits) that are conventionally displayed as ASN:ID (for example 64512:100)
- This notation however must be enabled using the **ip bgp-community new-format** command
- This community "tag" is stripped by neighbors by default (even within the same AS). Use neighbor statement with route-maps
- Routes can be in multiple communities

**MULTI EXIT_DISC/MED/METRIC Attribute**

This is an optional, non-transitive attribute meaning that its not required to pass on when received from another AS

MED can actually be transitive but only to one neighbor AS, not more

- Lower value preferred, but the default value is missing in IOS (treated as 0 which is equal and thus best)
    - This can be modified so that 0 is treated as worst by using the **bgp bestpath med missing-as-worst** command
- Influences preferred "exit point" when peering with the same AS at multiple locations
- Requirements are that WEIGHT, LOCAL_PREF, AS_PATH length and peer AS must be the same

- These requirements can also be modified, using the **bgp always-compare-med** command

Can be modified (increased from 0) using route-maps or redistribution

- Setting the MED outbound will influence the remote AS
- Setting the MED inbound will influence own AS

**ORIGINATOR_ID**

Route reflectors use the ORIGINATOR_ID and CLUSTER_LIST attributes for loop prevention

- Each routing table entry will have the originating routers RID inserted by the RR
- The RR will pass along routes to the RR-Clients, but it is these clients themselves who check the update for the presence of the RID (ORIGINATOR_ID)
    - If this ORIGINATOR_ID matches their own, the update is denied

**CLUSTER_LIST**

Set by RR by default to the value of the RID. The CLUSTER_ID identifies a group, or a single RR

- RR can use this information to discover other RR present on the network
- Prevents the installation of multiple routes in the BGP table that were reflected by RR neighbor

# Advertise-Map

## BGP Advertise-Map

Advertise prefixes based on the existence of other prefixes in the BGP table

- Network in the non-exist map has to be present in the BGP table
- Possible to advertise default route if another route is present
- This does not work with the **default-information originate** command

```
route-map NON_EXIST permit 10
 match ip address prefix-list Lo2
route-map ADV_MAP permit 10
 match ip address prefix-list ADV_PREFIX

ip prefix-list ADV_PREFIX permit 192.168.0.1/32
ip prefix-list Lo2 permit 192.168.0.2/32

router bgp 1
 address-family ipv4
  network 192.168.0.1 mask 255.255.255.255
  neighbor 10.0.13.3 advertise-map ADV_MAP non-exist-map NON_EXIST

show ip bgp neighbors 10.0.13.3 | i Condition
```

**Advertise default route**

```
route-map EXIST permit 10
 match ip address prefix-list Lo2
route-map ADV_MAP permit 10
```

```
 match ip address prefix-list ADV_DEFAULT

ip prefix-list ADV_DEFAULT permit 0.0.0.0/0
ip prefix-list Lo2 permit 192.168.0.2/32

ip route 0.0.0.0 0.0.0.0 null0

router bgp 1
 address-family ipv4
  network 0.0.0.0 mask 0.0.0.0
  neighbor 10.0.13.3 advertise-map ADV_MAP exist-map EXIST
```

# Aggregation

## ATOMIC_AGGREGATE Attribute

- Well-known discretionary attribute
- Must be recognized by all BGP implementations, but does not have to appear in all UPDATES
- This attribute is set when routes are aggregated (summarized) at a BGP speaker and forwarded to another AS
- Alerts BGP peers that information may have been lost in the aggregation and that it might not be the best path to the destination
- For example, if 10.0.1.0/24 and 10.0.2.0/24 are summarized to 10.0.0.0/22 it will include the 10.0.0.0/24 and 10.0.3.0/24
- Aggregated routes will inherit the attributes of the component routes
- When routes are aggregated, the aggregator attaches its RID to the route into the AGGREGATOR_ID attribute, unless the AS_PATH is set using the **as-set** statement
- It is not possible to disable the discard route in BGP

The **as-set** keyword preserves the AS_PATH information, meaning that the AS information is not overwritten by the aggregator

- In this case the AT_AGGREGATE is not set and the original AS (or multiple AS) is still present in the aggregated route
- The **as-confed-set** keyword is the same as **as-set**, but applies to confederations

By default the more specific routes are still advertised to peers in addition to the summary

- This behavior can be altered with the summary-only command to not advertise the more specific routes

## BGP Aggregation Suppress-Map

- Accomplishes the same as summary-only, except only subnets matched in the suppress-map will not be advertised

```
ip prefix-list 1 permit 3.0.0.1/32
ip prefix-list 3 permit 3.0.0.3/32
```

```
route-map SUPPRESS permit 10
 match ip address prefix-list 1
 match ip address prefix-list 3

router bgp 1
 address-family ipv4
  aggregate-address 3.0.0.0 255.255.255.252 as-set suppress-map SUPPRESS
```

## BGP Aggregation Unsuppress-Map

- Subnets matched will be advertised alongside the suppressed summary
- Applied on neighbor statement instead of aggregate statement
- Will not inherit attributes set by other neighbor statements such as community values
- In order to send attributes, neighbor route-map configurations have to be copied to the unsuppress route-map

```
ip prefix-list 1 permit 3.0.0.1/32
ip prefix-list 3 permit 3.0.0.3/32

route-map UNSUPPRESS permit 10
 match ip address prefix-list 1
 match ip address prefix-list 3
 set community 1:1 additive

router bgp 1
 address-family ipv4
  aggregate-address 3.0.0.0 255.255.255.252 as-set summary-only
  neighbor 10.0.12.2 unsuppress-map UNSUPPRESS
```

## BGP Inject-Map

- Routers can conditionally inject a more specific route based on the presence of an aggregate in the BGP table
- The injected subnets must fall within the range of the aggregate
- Copy-attributes will also transfer AS information, otherwise origin will be incomplete
- Route-source must match the neighbor the aggregate was received from, NOT the originator

```
ip prefix-list R3 permit 10.0.13.3/32
ip prefix-list AGG permit 3.0.0.0/29
ip prefix-list INJECT permit 3.0.0.5/32
ip prefix-list INJECT permit 3.0.0.6/32
ip prefix-list INJECT permit 3.0.0.7/32

route-map R3_AGG
 match ip address prefix-list AGG
 match ip route-source prefix-list R3
```

```
route-map INJECT
 set ip address prefix-list INJECT

router bgp 1
 address-family ipv4
  bgp inject-map INJECT exist-map R3_AGG copy-attributes

show ip bgp injected-paths
```

## BGP Aggregation Advertise-map

- This function works alongside **AS_SET** and **summary-only**
- Used when multiple AS share the same prefixes and are both included in the aggregation
- AS that are filtered do not become unreachable, they are just hidden in the aggregate

```
show ip bgp regexp _4$
ip as-path access-list 4 permit _4$

route-map AGG_HIDE_AS4 deny 10
 match as-path 4
route-map AGG_HIDE_AS4 permit 99

router bgp 1
 add ipv4
 aggregate-address 34.0.0.0 255.255.255.248 summary-only as-set advertise-map AGG_HIDE_AS4
```

## BGP Aggregation Attribute-Map

- Optionally add additional attributes to the aggregate with the **attribute-map** keyword
- Using a **route-map** instead will provide the same results, and will be converted to attribute-map in the config

```
route-map AGG_ATTRIBUTE permit 10
 set metric 500
 set local-preference 200
 set origin igp
 set community 3:1
 etc..

router bgp 1
 address-family ipv4
  aggregate-address 3.0.0.0 255.255.255.252 attribute-map AGG_ATTRIBUTE
  aggregate-address 3.0.0.0 255.255.255.252 route-map AGG_ATTRIBUTE
```

# AS_PATH

## BGP Prepend AS_PATH

**Prepend AS 3 five times to the AS_PATH (total of six entries in show ip bgp)**

```
route-map PREPEND_AS permit 10
 set as-path prepend 3 3 3 3 3

router bgp 1
 neighbor 10.0.12.2 route-map PREPEND_AS out

Prepend the last AS in the path 4 times. This will lead to 5 entries on neighbors (1 original + 4 prepend)
route-map PREPEND_LAST_AS permit 10
 set as-path prepend last-as 4

router bgp 1
 neighbor 10.0.12.2 route-map PREPEND_LAST_AS out
```

## BGP AS_PATH Tagging

- The AS_PATH is converted to a tag by default when redistributing BGP into IGP.
- Convert this tag back when redistributing the IGP back into another BGP process on another router with **set as-path tag**
- Configure **set automatic-tag** on the original redistributing router (BGP->OSPF) to preserve the origin code as well (i instead of ?)
- The automatic tag has to match a specific AS in the route-map
- It is not possible to prepend these redistributed routes using the same route-map

**AS_PATH tag**

```
route-map AS_PATH_TAG permit 10
 set as-path tag

router bgp 1
 add ipv4
  redistribute ospf 1 route-map AS_PATH_TAG
```

**Automatic Tag**

```
ip as-path access-list 1 permit .*
route-map AS_ORIGIN_TABLE_MAP permit 10
 match as-path 1
 set automatic-tag

router bgp 2
 add ipv4
  table-map AS_ORIGIN_TABLE_MAP

clear ip bgp ipv4 unicast table-map
```

## BGP Local-AS

- Use a different AS then is configured when neighboring with peers

By default R2 will see AS 2 prepended before the AS 1 on routes received from R1

- AS 2 will also be prepended to all R1 routes passed on to other BGP peers
- To override this behavior, configure **no-prepend** on R2

By default R1 will see AS 2 prepended before the actual AS of 64512

- To override this behavior, configure **replace-as** on R2
- R1 will see prefixes from R2 with AS2, other peers will see AS 64512

The **dual-as** keyword will allow R1 to peer with either the correct AS 64512 or the local-as 2.

```
router bgp 1
 neighbor 10.0.12.2 remote-as 2

router bgp 64512
 neighbor 10.0.12.1 remote-as 1
 neighbor 10.0.12.1 local-as 2 no-prepend replace-as dual-as
```

# Communities

## Communities Attribute

- This is an optional, transitive attribute that is comparable to route-tags
- Not used for best-path selection
- Large number (32 bits) that are conventionally displayed as ASN:ID (for example 64512:100)
- This notation however must be enabled using the ip bgp-community new-format command
- This community "tag" is stripped by neighbors by default (even within the same AS). Use neighbor statement with route-maps
- Routes can be in multiple communities

## BGP Filtering using Communities

There are three well-known BGP communities:

- No-advertise. Do not advertise route to any peers. CxFFFFFF02
- No-export. Do not advertise route to external peers. Advertise to internal and confederation peers only. OxFFFFF01
- Local-as. Do not advertise route to external and confederation peers. Advertise to internal peers only. OxFFFFFF03

It is possible to add the communities at the network statement or redistribution point.

- This will allow the router to advertise routes to specific neighbors only
- With the **neighbor** statement the routes are advertised to that neighbor but the community is applied after reception
- With the **network** statement the community is applied immediately

**Advertise routes with incomplete (?) origin (alternative to redistribution)**

```
route-map BGP_ORIGIN permit 10
 set origin incomplete

router bgp 1
 address-family ipv4
  network 1.0.1.0 mask 255.255.255.0 route-map BGP_ORIGIN
  network 1.0.2.0 mask 255.255.255.0 route-map BGP_ORIGIN
  network 1.0.3.0 mask 255.255.255.0 route-map BGP_ORIGIN
  network 1.0.4.0 mask 255.255.255.0 route-map BGP_ORIGIN

ip prefix-list Lo1 permit 1.0.1.0/24
ip prefix-list Lo2 permit 1.0.2.0/24
ip prefix-list Lo3 permit 1.0.3.0/24
ip prefix-list Lo4 permit 1.0.4.0/24

route-map BGP_COMMUNITY permit 10
 match ip address prefix-list Lo1
 set community no-advertise 1:1

route-map BGP_COMMUNITY permit 20
 match ip address prefix-list Lo2
 set community local-as 1:1

route-map BGP_COMMUNITY permit 30
 match ip address prefix-list Lo3
 set community no-export 1:1

route-map BGP_COMMUNITY permit 99

router bgp 1
 address-family ipv4
  neighbor 10.0.12.2 route-map BGP_COMMUNITY out

show ip bgp community 1:1
```

Match community values and modify on other routers:

- The set comm-list 1 delete keyword will only delete the community matched in the list
- The additive keyword will add the 'internet' community to the existing communities, and not overwrite them

```
ip community-list 1 permit no-export
ip prefix-list Lo4 permit 1.0.4.0/24

route-map MODIFY_COMMUNITY permit 10
 match ip address prefix-list Lo4
```

```
 match community 1
 set comm-list 1 delete
 set community internet additive
route-map RM1 permit 100

router bgp 2
 address-family ipv4
  neighbor 10.0.12.1 route-map MODIFY_COMMUNITY in
```

# iBGP Confederations

## iBGP Confederations

- Divides autonomous system into smaller sub-autonomous systems.
- Large (container) AS is a confederation, smaller systems are "members"
- Inside the member, regular iBGP rules apply using Full-mesh and/or RR
- Perceived externally only as the confederation AS
- Local Preference (LOCAL_PREF) Applies to the entire confederation

Configuration Considerations

- All routers must be configured to be aware of the confederation identifier
- All routers should be configured to be aware of all the confederation peers (members)
- Can be configured alongside RR

```
router bgp 1
 bgp confederation identifier 123
 bgp confederation peers 2
 bgp confederation peers 3
neighbor 192.168.0.2 remote-as 2
neighbor 192.168.0.3 remote-as 3
neighbor 192.168.0.2 update-source Lo0
neighbor 192.168.0.3 update-source Lo0
neighbor 192.168.0.2 ebgp-multihop
neighbor 192.168.0.3 ebgp-multihop
 address-family ipv4
  neighbor 192.168.0.2 next-hop-self
  neighbor 192.168.0.3 next-hop-self
```

The peering between AS1 and AS2, AS3 is an internal peering using different AS

- Meaning that routers R3 and R2 are unaware of the routes injected by R1, unless a full-mesh is configured
- Another option is to use **next-hop-self** for all peers

When peering with loopbacks between confederation peers the peering is basically the same as eBGP

- Meaning that **ebgp-multihop** has to be configured

# eBGP Peering

## eBGP Peering Rules

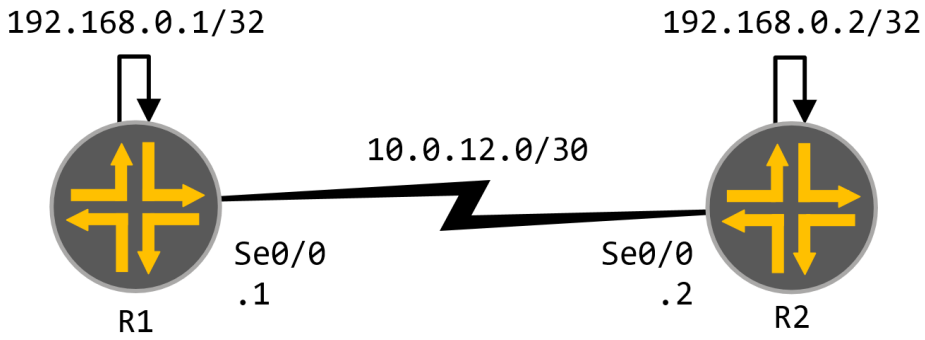
- Neighbor must be in a different AS
- Neighbor should be directly connected. Override with **ebgp-multihop** (or **ttl-security hops**) which changes the default TTL
- If not, an underlying routing protocol must provide reachability (static, IGP). Beware of recursive routing (see below)
- NEXT_HOP is changed (to the update-source) when routes are advertised (by the advertising router)
- Routes with own AS are rejected if observed in incoming AS_PATH updates (AS_SEQUENCE and / or AS_SET)
- All the best routes (and only the best routes) are advertised and accepted in the UPDATE message

## Recursive Routing Problems & Backdoor Statement

- If using an IGP to advertise loopbacks for the external peering (multi-hop). It is important not to advertise the same loopbacks into BGP
- BGP has a lower AD, meaning that the IGP routes used for the peering will become BGP routes. This will lead to recursive routing
- Solve by increasing the AD of eBGP (not recommended), decreasing the AD of the IGP (possible solution) or use backdoor routes

BGP Backdoor statement

- The **backdoor** statement gets added to the network command (**network x.x.x.x mask x.x.x.x backdoor**)
- This command does NOT advertise a route into bgp (even though the **network** statement is being used)
    - Instead this command tells the BGP process to change the distance of that specific route to iBGP when it is received from a eBGP neighbor
    - BGP will install the route in the routing table, but instead of a distance of 20 (eBGP) it will give it a distance of 200 (iBGP)
    - For this reason you add the **backdoor** statement to the route you RECEIVE from a neighbor, not your own route



**R1**

```
int se0/0
 ip add 10.0.12.1 255.255.255.252
interface lo0
 ip address 192.168.0.1 255.255.255.255

router eigrp 1
 network 10.0.12.1 0.0.0.0
 network 192.168.0.1 0.0.0.0

router bgp 1
 neighbor 192.168.0.2 remote-as 2
 neighbor 192.168.0.2 ebgp-multihop 2
 neighbor 192.168.0.2 update-source Loopback0
 address-family ipv4
  network 192.168.0.1 mask 255.255.255.255
  network 192.168.0.2 mask 255.255.255.255 backdoor
```

**R2**

```
int se0/0
 ip add 10.0.12.2 255.255.255.252
interface lo0
 ip address 192.168.0.2 255.255.255.255

router eigrp 1
 network 10.0.12.2 0.0.0.0
 network 192.168.0.2 0.0.0.0

router bgp 2
 neighbor 192.168.0.1 remote-as 1
 neighbor 192.168.0.1 ebgp-multihop 2
 neighbor 192.168.0.1 update-source Loopback0
 address-family ipv4
  network 192.168.0.1 mask 255.255.255.255 backdoor
  network 192.168.0.2 mask 255.255.255.255
```

## eBGP Peering using Default Route

BGP will not actively open (initiate) BGP session with a peer if the only route to it is the default route

- BGP will initiate a session if the active BGP peer has a more specific route to its peer
- The passive BGP peer is allowed to only have a default route
- One side has to have a specific route for this to work

**Force active/passive peerings between peers**

```
router bgp 1
 neighbor 192.168.0.2 transport connection-mode active
```

```
router bgp 2
 neighbor 192.168.0.1 transport connection-mode passive
```

If Peer 1 (being the active peer by force) only has a default route to Peer 2, the session will not form

```
ip route 0.0.0.0 0.0.0.0 10.0.12.2

ip bgp-community new-format
router bgp 1
 bgp router-id 192.168.0.1
 neighbor 192.168.0.2 remote-as 2
 neighbor 192.168.0.2 ebgp-multihop
 neighbor 192.168.0.2 update-source Loopback0
 neighbor 192.168.0.2 transport connection-mode passive
 address-family ipv4
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 send-community
```

### BGP Connected-Check

- Allows directly connected neighbors to peer with loopback addresses without configuring multi-hop

```
router bgp 1
 neighbor 10.0.12.2 remote-as 2
 neighbor 10.0.12.2 disable-connected-check
```

# iBGP

### BGP Internal Routing

- Does not replace IGP. No direct connectivity required
- Works alongside IGPs, this is why direct connectivity is not required
- Carry external prefixes throughout the AS
- "Split-horizon" as the loop-prevention. iBGP routes are not forwarded to other iBGP peers
- Does not re-advertise internally-learned (iBGP) prefixes to other iBGP peers, because the AS_PATH is not modified inside the AS
- Only advertises the best route in any BGP UPDATE
- Multiple sessions to same neighbor are not permitted, use loopback destinations instead

Peering Rules

- Neighbor must be in the same AS
- Neighbors do not have to be directly connected, if not a underlying routing protocol must provide reachability

Default Policy Behavior

- NEXT_HOP is not changed when routes are advertised
- External routes are advertised
- Routes learned from other internal peers are not advertised to other iBGP neighbors. But are advertised to eBGP neighbors

## iBGP Next-Hop-Self

- Because NEXT_HOP is not changed, IGPs are needed to reach neighbors
- Another solution is to change the NEXT_HOP to the local router address
- The address that will be chosen for this is the loopback address (in case of next-hop-self) or the internal address used to peer with the neighbors
- Make sure the networks next_hop_self is set to are advertised into iBGP. Another solution is to redistribute the IGP

```
router bgp 1
 neighbor 10.0.12.2 remote-as 2
 address-family ipv4
  neighbor 10.0.12.2 next-hop-self all
  network 10.0.12.0 mask 255.255.255.0
  network 10.0.13.0 mask 255.255.255.0
  redistribute ospf 1 metric 2 match internal
```

The all keyword enables next-hop-self for both eBGP and iBGP received paths. Default is only for iBGP received paths

- BGP only redistributes ebgp routes into an IGP
- BGP does not redistribute ospf external routes by default
- If no metric is specified it will be equal to the number of hops a peer needs to reach the networks
- If a metric is specified, it will be applied to all routes advertised, meaning that all routes in the OSPF domain will receive the same metric
- The default is to only match OSPF internal routes, redistributed routes into OSPF are not included

**iBGP Peer-Groups**

```
router bgp 123
 bgp listen range 10.0.123.0/24 peer-group PEERS
 bgp listen limit 2
 neighbor PEERS peer-group
 neighbor PEERS remote-as 123
 address-family ipv4
  neighbor PEERS activate
  neighbor PEERS send-community
  neighbor PEERS etc..
```

# IPv6

## IPv6 Networks over IPv4

```
router bgp 1
 neighbor 10.0.12.2 remote-as 2
 address-family ipv4
  neighbor 10.0.12.2 activate
 address-family ipv6
  neighbor 10.0.12.2 activate
  neighbor 10.0.12.2 route-map IPV6_NEXT_HOP out
  network 1::1/128

route-map IPV6_NEXT_HOP permit 10
 set ipv6 next-hop 2001:10:0:12::1
```

## IPv6 Peer Link-Local

```
router bgp 1
  neighbor FE80::2%Serial1/0 remote-as 2
 address-family ipv6
  neighbor FE80::2%Serial1/0 activate
  network 1::1/128
```

## IPv6 Peering using Loopbacks

- Same as IPv4, IPv6 neighbors are not automatically activated
- Specify router-id if no IPv4 addresses are used on the router

```
ipv6 route 2::2/128 2001:10:0:12::2

router bgp 1
 bgp router-id 192.168.0.1
 neighbor 2::2 remote-as 2
 neighbor 2::2 update-source Loopback 0
 neighbor 2::2 disable-connected-check
 address-family ipv6
  neighbor 2::2 activate
```

## IPv4 Networks over IPv6

- Peer with directly connected interfaces (NOT loopbacks) when advertising IPv4 prefixes over an IPv6 connection

```
router bgp 1
 neighbor 2001:10:0:12::2 remote-as 2
 address-family ipv4
  neighbor 2001:10:0:12::2 activate
  neighbor 2001:10:0:12::2 route-map IPV4_NEXT_HOP in
  network 192.168.0.1 mask 255.255.255.255
```

```
address-family ipv6
 neighbor 2001:10:0:12::2 activate

route—map IPV4_NEXT_HOP permit 10
 set ip next—hop 10.0.12.2
```

# Filtering AS

## AS-Path Filters

| | |
|---|---|
| _ | White space, start of string or end of string. |
| ^ | Start of string, PEER AS. |
| $ | End of string, ORIGINATING AS. |
| . | Matches any character. |
| + | repeats the previous character one or more times. |
| * | repeats the previous character zero or many times |
| ? | repeats the previous character one or zero times. |
| I | A logical "or" statement. |
| \ | Removes special meanings. |
| () | Matches a group of characters. |
| [] | Matches a range of characters. |

Definition and Use of AS-Path Access-Lists

- AS-Path access-lists match a single character, every number in an AS is a single character.
  - Example. AS 2456 consists of the single characters 2,4,5 and 6.
- Can match a range of characters using brackets [].
  - Example. [0-3] consists of 0 or 1 or 2 or 3
- Can match a group of characters.
  - Example. (123) matches only 123 when the characters follow each other. However they can be part of a greater AS such as 55123 or 12300.
  - Example. (123_) matches only 123 when its at the end of the AS, such as 55123 but not 12300.
  - Example. (_123_) matches only AS123 and it cannot be part of a greater AS.

## AS-Path AS-Sequence

- In IOS the AS-Sequence is the AS_PATH that a route travels through
- The AS that the router receives the route from is called the PEER AS
- The router that originated the route is called the ORIGIN AS, others are called TRANSIT AS

- Example. ^500_400_300_200_100$. 500 is the PEER AS, 100 is the ORIGIN AS, others are TRANSIT AS

## AS-Path Filtering

**Match prefixes that originated in the connected AS**

```
ip as-path access-list 1 permit ^[0-9]+$
route-map BGP_CONNECTED_AS permit 10
 match as-path 1

router bgp 1
 address-family ipv4
  neighbor 10.0.12.2 route-map BGP_CONNECTED_AS in
  neighbor 10.0.12.2 filter-list 1 in
```

- This matches all numbers but does not allow blank spaces. Meaning that there can only be one AS in the path, which is the neighbors AS
- The + means that the pattern must appear, so blank AS (our own) will not match

| | |
|---|---|
| ^$ | Local AS |
| .* | All AS |
| ^5_ | Directly connected AS 5 |
| _5_ | Transferring AS 5 |
| _5$ | Originated in AS 5 |
| ^[0-9]+$ | Originated in directly connected AS |
| ^[0-9]*$ | Originated in directly connected AS + empty AS (local AS) |
| ^([0-9]+)_5 | AS 5 which passed through directly connected AS |
| ^2_([0-9]+) | Directly connected to AS 2 |
| ^\(1234\) | Confederation peer 1234. |
| _(4\|5)$ | Originated in AS 4 or AS 5 |
| ^(2\|3)$ | Originated in AS 2 or AS 3 that are directly connected |
| _2_(4\|5)$ | Originated in AS 4 or AS 5 that passed through AS 2 |
| ^[0-9]+([0-9]+)?$ | Originated in directly connected AS or directly connected to our directly connected AS |

? Basically means true or false, the secondary AS that are being matched can appear or not

# Filtering PL / ACL

## BGP Extended Access-Lists Filtering

```
ip access-list extended PREFIXES
 deny ip 2.0.0.0 0.0.0.255 host 255.255.255.252
 deny ip 2.0.0.0 0.0.0.255 host 255.255.255.254
 permit ip any any

router bgp 1
 address-family ipv4
  neighbor 10.0.12.2 distribute-list PREFIXES in
```

## BGP Prefix-List Filtering

```
ip prefix-list PREFIXES deny 1.0.0.2/31
ip prefix-list PREFIXES deny 1.0.0.4/30
ip prefix-list PREFIXES permit 0.0.0.0/0 le 32

router bgp 2
 address-family ipv4
  neighbor 10.0.12.1 prefix-list PREFIXES in
```

## BGP Outbound Route Filtering (ORF)

- Normally when a router filters incoming routes the peer will still advertise them.
- To signal the peer that some routes do not need to be advertised, configure an ORF based on the same prefix-list that is filtering.

```
ip prefix-list PREFIXES deny 1.0.0.1/32
ip prefix-list PREFIXES deny 1.0.0.2/32
ip prefix-list PREFIXES deny 1.0.0.3/32
ip prefix-list PREFIXES deny 1.0.0.4/32
ip prefix-list PREFIXES permit 0.0.0.0/0 le 32

router bgp 2
 neighbor 10.0.12.1 remote-as 1
 address-family ipv4
  network 192.168.0.2 m 255.255.255.255
  neighbor 10.0.12.1 activate
  neighbor 10.0.12.1 capability orf prefix-list send
  neighbor 10.0.12.1 prefix-list PREFIXES in

router bgp 1
 neighbor 10.0.12.2 remote-as 2
 address-family ipv4
  neighbor 10.0.12.2 activate
```

```
 neighbor 10.0.12.2 capability orf prefix-list receive
 network 192.168.0.1 m 255.255.255.255
 network 1.0.0.1 m 255.255.255.255
 network 1.0.0.2 m 255.255.255.255
 network 1.0.0.3 m 255.255.255.255
 network 1.0.0.4 m 255.255.255.255

show ip bgp neighbors 10.0.12.1 | s capabilities
show ip bgp neighbors 10.0.12.2 advertised-routes
```

# Misc

## BGP Authentication

- BGP only supports one type of authentication, MD5
- Configure authentication using the **neighbor x.x.x.x password** command

## BGP Fall-over

- Holdtime does not have to expire in order to tear down the session.
- The fall-over will still allow bgp to form neighborships using a default or summary route.
- Configure a route-map to make sure that the specific route has to be present in the routing table.

```
ip route 192.168.0.2 255.255.255.255 10.0.12.2

ip prefix-list R2_LOOPBACK permit 192.168.0.2/32
route-map R2_FALLOVER permit 10
 match ip address prefix-list R2_LOOPBACK

router bgp 1
 neighbor 192.168.0.2 fall-over route-map R2_FALLOVER
```

## BGP Next Hop Tracking (NHT)

- On-by-default feature that notifies BGP to a change in routing for BGP prefix next-hops.
- NHT makes the process of dealing with next-hop changes event-driven, instead of using the 60 second scanner process.
- The bgp nexthop trigger delay defines how long for the NHT process to delay updating BGP.

```
router bgp 1
 bgp nexthop trigger enable
 bgp nexthop trigger delay 5
```

## BGP Selective Address Tracking (SAT)

- The route-map determines what prefixes can be seen as valid prefixes for next-hops.

- Allows for specific addresses as viable next-hops, and will pull prefixes from the bgp table if the next-hop does not match.
- Re-use the same route-map in fall-over to also tear down the session.

```
ip prefix-list LOOPBACKS permit 0.0.0.0/0 ge 29
route-map SAT permit 10
 match ip address prefix-list LOOPBACKS

router bgp 1
 address-family ipv4
  bgp nexthop route-map SAT
  neighbor 10.0.13.3 fall-over route-map SAT
  neighbor 10.0.12.2 fall-over route-map SAT
```

## BGP Multi-Session

**Use a different TCP session for each address-family**

```
router bgp 1
 neighbor 192.168.0.2 transport multi-session (must agree on both sides)
```

## BGP Peer-Groups

- Applying filters and other settings on a peer-by-peer basis can result in high CPU usage
- Peer groups allow you to logically group peers together and apply policies to the group, this can dramatically reduce CPU load
- Routers will still send out BGP UPDATE messages on a peer-by-peer basis (they are not broadcasted/multicasted), this is due to the nature of BGP (individual TCP sessions)

```
ip prefix-list ROUTE_FILTER deny 10.0.0.0/8 le 32
ip prefix-list ROUTE_FILTER deny 172.16.0.0/12 le 32
ip prefix-list ROUTE_FILTER deny 192.168.0.0/16 le 32
ip prefix-list ROUTE_FILTER permit 0.0.0.0/0 le 32

router bgp 1
 neighbor 10.0.12.2 remote-as 2
 neighbor 10.0.13.3 remote-as 3
 neighbor PEERS peer-group
 address-family ipv4
  neighbor PEERS activate
  neighbor PEERS send-community
  neighbor PEERS prefix-list ROUTE_FILTER in
```

## BGP TTL-Security

- Configuring TTL-Security automatically allows peering over multiple hops (without explicitly configuring multi-hop).

- Configuring **ttl-security hops 2** will cause the router to expect incoming packets have a TTL value of 253.
- If packets arrive with a TTL of lower than 253, they will be discarded.

```
router bgp 1
 bgp router-id 192.168.0.1
 neighbor 192.168.0.2 remote-as 2
 neighbor 192.168.0.2 update-source Loopback0
 neighbor 192.168.0.2 ttl-security hops 2
 address-family ipv4
  neighbor 192.168.0.2 activate
```

# Route-Reflector (RR)

## iBGP Route-Reflector

- Because routes learned from other internal peers are not advertised to other iBGP neighbors, iBGP needs a full-mesh topology.
- This approach has drawbacks however when a lot of routers are present in the AS.
- Another solution is to use a Route-Reflector (RR) to basically re-advertise the routes to internal peers.

**BGP Route-Reflectors Terminology**

- Route-Reflector (RR), the router that has to feature enabled on some or all interfaces.
- Route-Reflector Client, router peering with the RR for which the RR has the feature enabled.
- Non-Client, router that is in the same AS but is not enabled for the feature on the RR.

**A Route Reflector reflects routes considered as best routes only**

- If more than one update is received for the same destination, only the BGP best route is reflected.
- A Route Reflector is not allowed to change attributes of the reflected routes including the next-hop attribute.

**Route Target Constraint**

With Route Target Constraint (RTC) the Route Reflector sends only wanted VPN4/6 prefixes to the PE.

- 'Wanted' means that the PE has a VRF importing the specific prefixes.

**ORIGINATOR_ID**

Route reflectors use the ORIGINATOR_ID and CLUSTER_LIST attributes for loop prevention.

- Each routing table entry will have the originating routers RID inserted by the RR.
- The RR will pass along routes to the RR-Clients, but it is these clients themselves who check the update for the presence of the RID (ORIGINATOR_ID). If this ORIGINATOR_ID matches their own, the update is denied.

**CLUSTER_LIST**

Set by RR by default to the value of the RID. The CLUSTER_ID identifies a group, or a single RR.

- RR can use this information to discover other RR present on the network.

- Prevents the installation of multiple routes in the BGP table that were reflected by RR neighbors.

Set the CLUSTER_ID to be equal on all reflectors

```
router bgp 1
 bgp cluster-id 13.13.13.13
```

## Advertisement Rules

- Routes received from iBGP are advertised to external peers.
- So there is no need for RR in this case.

Non-client -> iBGP = Forwarded to eBGP and RR-Clients.

- When a NC receives a route from an eBGP peer it will be forwarded to the RR, which forwards it to all eBGP peers and iBGP RR-Clients.

External route from RR-Client -> iBGP = Forwarded to eBGP, RR-Clients and non-clients

- When a RR-Client receives a route from an eBGP peer it will be forwarded to the RR, which forwards it to all eBGP peers and iBGP peers.

External route from RR -> iBGP = Forwarded to eBGP, RR-Clients and non-clients

- When a RR receives a route from an eBGP peer it will be forwarded to all eBGP and iBGP peers.

# Session

## BGP Session

- BGP uses TCP port 179 for destination and a random source port.
- Commands that affect the BGP session are: shutdown, password, update-source and modification of timers.
- When using the network statement, BGP looks in the routing table and verifies that the route exists.
- The subnet used in the network statement must match exactly in order to inserted into BGP.

If two BGP peers initiate a session, the router that initiated the session will use a random TCP port to set up the TCP connection, the router that received the session request will use TCP port 179.

- The initiating router message will be OPEN_ACTIVE.
- The receiving router message will be OPEN_PASSIVE.
- The receiving router will also send the OPEN_ACTIVE message, but this will fail because it is no longer needed.

**BGP Messages**

| | |
|---|---|
| OPEN | First message sent between peers after a TCP connection has been established<br>Exchanges parameters (ASN / authentication / capabilities / etc.)<br>Confirmed by KEEPALIVE message or denied by NOTIFICATION message |
| KEEPALIVE | Sent periodically to ensure that the connection is live |

| | |
|---|---|
| | If no KEEPALIVE message is received within the negotiated hold-timer, the session is torn down |
| NOTIFICATION | Sent in response to errors or special conditions<br>If a connection encounters an error, a NOTIFICATION message is sent and the connection is closed |
| UPDATE | Exchanges Path Attributes (PAs) with associated prefixes/NLRI information (routes)<br>Multiple prefixes that have the same PAs can be advertised in a single message<br>For example AS_PATH 1 2 3 can be associated with 100 prefixes advertised in a single UPDATE message<br>A single UPDATE message may simultaneously advertise a feasible prefix and withdraw multiple unfeasible prefixes from service |

When BGP fast peering session deactivation is enabled, BGP will monitor the peering session with the specified neighbor.

- Adjacency changes are detected, and terminated peering sessions are deactivated in between the default or configured BGP scanning interval.

**BGP States**

| **STATE** | **DESCRIPTION / REASON STUCK IN STATE** | **RESULT** |
|---|---|---|
| IDLE | No peering, router is looking for neighbor<br>Connections refused | CONNECT if successful |
| IDLE (admin) | administratively shut down (neighbor x.x.x.x shutdown) | - |
| CONNECT | Establishing TCP session | ACTIVE if successful |
| ACTIVE | TCP session connection is established<br>No messages have been sent or received yet | OPENSENT if successful |
| OPENSENT | OPEN message is sent to peer<br>Router is OPEN and expects an OPEN message from peer<br>Peer OPEN message has not yet been received | OPENCONFIRM if successful |
| OPENCONFIRM | An OPEN message is both sent and received<br>Leads to either KEEPALIVE (approved) or NOTIFICATION (not approved) | KEEPALIVE if all parameters match / session approved<br>NOTIFICATION if mismatch between parameters / session denied |
| ESTABLISHED | Routers have a BGP peering session | UPDATES messages are exchanged between BGP speakers. |

```
R1#show ip bgp summary
BGP router identifier 192.168.0.1, local AS number 1
BGP table version is 9, main routing table version 9
4 network entries using 592 bytes of memory
4 path entries using 256 bytes of memory
2/2 BGP path/bestpath attribute entries using 272 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1144 total bytes of memory
BGP activity 10/6 prefixes, 12/8 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.13.3       4            1      12      17        9    0    0 00:03:36        0
192.168.0.2     4            2      15      14        9    0    0 00:03:23        2
```

- The neighbor address is the configured address (neighbor statement), not necessarily the neighbor's router-id
- Any number in the State/PfxRcd column means that the state is established
- BGP neighbors exchange Path Attributes (PA) (AS_PATH etc.) that come with the prefix information (subnets), not the other way around

## Clearing the BGP Neighborship

| Command | Reset Type | Direction | Route-Refresh | Soft-Reconfiguration Inbound |
|---|---|---|---|---|
| clear ip bgp * | hard | both | - | - |
| clear ip bgp * out | soft | out | X | - |
| clear ip bgp * in | soft | in | X | - |
| clear ip bgp * soft out | soft | out | - | X |
| clear ip bgp * soft in | soft | in | - | X |
| clear ip bgp * soft | soft | both | - | - |

- The reset type soft will not bring down the entire neighborship and will only change the BGP updates
    - When issued, these commands cause the router to reevaluate its existing BGP table and create a new BGP Update for that neighbor
- The '**soft**' version of the commands are the older style commands that require **soft-reconfiguration inbound** configured
- The 'non-soft' version of the commands are the newer style and work based on route-refresh capabilities (enabled by default)
    - Route-refresh allows the router to request a new full bgp update and apply its own inbound BGP filters afterwards

Clearing a neighbor (asking for a full update) using route-refresh will display new output in the **show ip bgp summary** command

- E - The peer has requested a route-refresh that hasn't been completed yet

- S - The refresh for that peer has started
- P - Displayed if BGP is refreshing the neighbor starting at some table version other than zero. It should normally start from zero so you won't see this displayed but if you have multiple peers that are out of sync and they request route-refresh at the same time then you may see the P displayed for them once they are in sync

You can see a combination of this output (SE for example) meaning that the refresh has started but not yet completed.

## BGP Output

```
R1#show ip bgp neighbors 192.168.0.2
BGP neighbor is 192.168.0.2,  remote AS 2, external link
  BGP version 4, remote router ID 192.168.0.2
  BGP state = Established, up for 00:11:20
  Last read 00:00:55, last write 00:00:22, hold time is 180, keepalive interval is 60 seconds
  >>
  Connections established 4; dropped 3
  Last reset 00:15:23, due to Peer closed the session of session 1
  External BGP neighbor may be up to 2 hops away.
  Transport(tcp) path-mtu-discovery is enabled
  Graceful-Restart is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 2
Local host: 192.168.0.1, Local port: 179
Foreign host: 192.168.0.2, Foreign port: 30689
Connection tableid (VRF): 0
Maximum output segment queue size: 50
```

- The external link means that this is an eBGP neighbor
- eBGP multihop is enabled (2 hops)
- The neighbor initiated the TCP session (port 30689)

```
R2#show ip route 1.0.0.1 255.255.255.255
Routing entry for 1.0.0.1/32
  Known via "bgp 2", distance 20, metric 0
  Tag 13, type external
  Last update from 10.0.12.1 00:13:15 ago
  Routing Descriptor Blocks:
  * 10.0.12.1, from 10.0.12.1, 00:13:15 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 13
      MPLS label: none
```

- R2 (AS1) is an eBGP neighbor with R2 at 10.0.12.2, R1 advertises 1.0.0.1/32 into BGP AS 13
- Known via bgp 2 means that R2 learned the route from its own AS2, the neighbor's AS (AS13) is not listed
- Type external means a eBGP route (another indicator is the distance, however this can be changed)

- 'from 10.0.12.1' is the neighbor that advertised the route to R2, in this case 10.0.12.1 is also the original originator of the route
- The route 1.0.0.1/32 is 1 AS hop away

```
R2#show ip route bgp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 4 subnets
B        1.0.0.1 [20/0] via 10.0.12.1, 00:30:41
B        1.0.0.2 [20/0] via 10.0.12.1, 00:30:14
B        1.0.0.3 [20/0] via 10.0.12.1, 00:30:14
B        1.0.0.4 [20/0] via 10.0.12.1, 00:30:14
```

- eBGP AD is 20 by default, iBGP is 200 by default
- The metric is always 0 unless MED is configured

```
R3#show ip route 10.0.20.0 255.255.255.0
Routing entry for 10.0.20.0/24
  Known via "bgp 1", distance 200, metric 0
  Tag 2, type internal
  Last update from 10.0.13.1 00:20:36 ago
  Routing Descriptor Blocks:
  * 10.0.13.1, from 10.0.13.1, 00:20:36 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 2
      MPLS label: none
```

- R3 (AS1) is an iBGP neighbor with R1 at 10.0.13.3
- R3 knows about the route via BGP 1 with type internal (iBGP) and distance 200
- The update is received from 10.0.13.1, not the originator of the route (R2 at 192.168.0.2)
- This route is known by R2 via 10.0.13.1 because **next-hop-self** was configured on R1

```
R6#show ip bgp 1.0.0.1/32
BGP routing table entry for 1.0.0.1/32, version 2
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 2
  2 13
    10.0.26.2 from 10.0.26.2 (192.168.0.2)
      Origin IGP, localpref 100, valid, external, best
```

- R6 learns the 1.0.0.1/32 route from neighbor 10.0.26.2
- The first 10.0.26.2 means the next-hop of this route

- 'from 10.0.26.2' means the neighbor that advertised the route in the update message
  - In eBGP neighborships this address can often be the same address because there is only a single peering
- 192.168.0.2 is the router-id of this peer
- The path attributes (AS_PATH) is AS2 (10.0.26.2) -> AS1 (10.0.12.1)
- External means that it's an eBGP route and that is best ( so both the > and i will be displayed for this route by the **show ip bgp command**)
- A next-hop with (inaccessible) means a routing issue, this is often the case when receiving iBGP routes when there is no next-hop-self or IGP configured

```
R2#show ip bgp neighbors 10.0.26.6 advertised-routes
BGP table version is 5, local router ID is 192.168.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.0.0.1/32       10.0.12.1                0             0 13 i
 *>  1.0.0.2/32       10.0.12.1                0             0 13 i
 *>  1.0.0.3/32       10.0.12.1                0             0 13 i
 *>  1.0.0.4/32       10.0.12.1                0             0 13 i
```

From R2's perspective the AS path to the 1.0.0.1-4/32 prefixes is AS 13. R6 will add the AS of the router it received the prefixes from to the AS_PATH (AS 13 2)

- Updates advertised to eBGP peers do not include the LOCAL_PREFERENCE PA

```
E2# show ip bgp 176.0.0.0/4 longer-prefixes
! legend omitted for brevity

   Network          Next Hop            Metric LocPrf Weight Path
*>i181.0.0.0/8      10.100.1.1               0    100      0 1 1811 i
*                   192.168.1.6              0             0 3 2 50 51 52 1811 i
*>i182.0.0.0/8      10.100.1.1               0    100      0 1 2 1822 i
*                   192.168.1.6              0             0 3 2 50 51 1822 i
* i183.0.0.0/8      10.100.1.1               0    100      0 1 2 50 1833 i
*>                  192.168.1.6              0             0 3 2 50 1833 i
*> 184.0.0.0/8      192.168.1.6              0             0 3 2 1844 i
*> 185.0.0.0/8      192.168.1.6              0             0 3 1855 i
```

- The local preference value is absent from eBGP learned routes
- The weight is always 0 for all learned routes (unless altered on the local router)
- Updates received from eBGP peers do not include the LOCAL_PREFERENCE PA
- IOS lists a null value for LOCAL_PREFERENCE for eBGP-learned routes by default

# Table-Map

## BGP Table-Map

- Sits between the BGP table and the RIB. Does not filter bgp prefixes received.
- Can be used to match prefixes and apply tags or communities.
- Can also be used to filter BGP prefixes from the local RIB with the filter keyword.
- Typically used to apply QoS settings and tags/communities.
- Applies to all prefixes received from all peers.

**Filter specific prefixes from the RIB**

```
ip prefix-list DENY_192 permit 192.168.0.2/32
route-map TABLE_MAP_DENY deny 10
 match ip add prefix DENY_192

router bgp 1
 add ipv4
  table-map TABLE_MAP_DENY filter

clear ip bgp ipv4 unicast table-map
```

**Apply QoS to specific communities**

```
ip community-list standard 2:2 permit 2:2
ip community-list standard 3:3 permit 3:3

route-map TABLE_MAP_QOS permit 10
 match community 2:2
 set ip precedence 2
route-map TABLE_MAP_QOS permit 20
 match community 3:3
 set ip precedence 3
route-map TABLE_MAP_QOS permit 99

router bgp 1
 add ipv4
  table-map TABLE_MAP_QOS
```

**Verify QoS application to routes**

```
show ip cef 192.168.0.1/32
```

**Configure the ingress interface to perform classification based on IPP**

```
int fa0/0
 bgp-policy source ip-prec-map
 bgp-policy destination ip-prec-map

BGP Table-Map Automatic-Tag
```

```
ip as-path access-list 1 permit .*
route-map AS_ORIGIN_TABLE_MAP permit 10
 match as-path 1
 set automatic-tag

router bgp 2
 add ipv4
  table-map AS_ORIGIN_TABLE_MAP

clear ip bgp ipv4 unicast table-map
```

# Cryptography

## Internet Key Exchange (IKE)

- IKEv1 uses UDP port 500. UDP port 4500 is used for NAT-Traversal in IKEv2.
- Encapsulation Header (ESP) is UDP port 50. Authentication Header (AH) is UDP port 51.

**IKE Phase 1 (P1)**

- IKE authenticates IPsec peers.
- Negotiates IKE Security Associations (SAs).
- Sets up a secure channel for negotiating IPsec SAs in P2.
- Protects the identities of IPsec peers.
- Main mode hides the identities of the two sides, but takes more time to negotiate. Default mode.
- Aggressive mode takes less time to negotiate at the cost of less security.
- MM and AG states differ, aggressive mode skips the SA_SETUP state. Both lead to QM_IDLE.

**IKE Phase 2 (P2)**

- IKE negotiates IPSec SA parameters.
- Sets up matching IPSec SAs in the peers.
- Sets up protection suite using ESP or AH.
- Two SAs are set up. One for sending and one for receiving traffic.
- Multiple P2 SAs can be established over the same P1 SA.
- Only one state, QM_IDLE.

**Main Mode States**

1. MM_NO_STATE.
2. MM_SA_SETUP. The peers have agreed on parameters for the ISAKMP SA.
3. MM_KEY_EXCH. Exchanged DH information but remains unauthenticated.
4. MM_KEY_AUTH. Authenticated.

**Aggressive Mode States**

1. AG_NO_STATE.
2. AG_INIT_EXCH. Exchanged DH information but remains unauthenticated.
3. AG_AUTH. Authenticated.

# Crypto Maps

## Crypto Maps

- Requires specification of traffic (ACL) in order to function.
- Applied on physical interface.
- The VPN will only be initiated when traffic is generated that matches the ACL.

```
show crypto map
show crypto isakmp profile

debug crypto isakmp
debug crypto ipsec
```

**Main Mode**

```
crypto isakmp policy 10
 encr aes 256
 hash sha256
 authentication pre-share
 group 14
 lifetime 28800

crypto ipsec transform-set TS esp-aes 256 esp-sha256-hmac
 mode tunnel

crypto isakmp key cisco address 10.0.12.2

crypto map CMAP 10 ipsec-isakmp
 set peer 10.0.12.2
 set transform-set TS
 match address VPN
 qos pre-classify

ip access-list extended VPN
 permit ip host 192.168.0.1 host 192.168.0.2

ip route 192.168.0.2 255.255.255.255 10.0.12.2

int fa0/0
 crypto map CMAP
```

**Aggressive Mode**

```
crypto isakmp policy 10
 encr aes 256
```

```
 hash sha256
 authentication pre-share
 group 14
 life time 3600

crypto ipsec transform-set TS esp-aes 256 esp-sha256-hmac
 mode tunnel

crypto keyring KEYRING
 pre-shared-key address 10.0.12.2 key cisco

crypto isakmp profile ISAKMP
 keyring KEYRING
 initiate mode aggressive
 match identity address 10.0.12.2

crypto map CMAP 10 ipsec-isakmp
 set peer 10.0.12.2
 set transform-set TS
 match address VPN
 qos pre-classify
 set isakmp ISAKMP

ip access-list extended VPN
 permit ip host 192.168.0.1 host 192.168.0.2

ip route 192.168.0.2 255.255.255.255 10.0.12.2

int fa0/0
 crypto map CMAP
```

# Dynamic VTI

## Dynamic Virtual Tunnel Interfaces (VTI)

- Provides a separate virtual interface for each VPN session cloned from virtual template.
- When using EIGRP the split-horizon and next-hop-self configuration needs to be placed on the virtual-template.
- The VPN will be initiated even if no traffic is generated.

**Hub configuration**

```
crypto isakmp policy 10
 encryption aes 256
 hash sha256
 authentication pre-share
 group 14
```

```
 lifetime 3600

crypto keyring KEYRING
 pre-shared-key address 123.0.0.2 key cisco
 pre-shared-key address 123.0.0.3 key cisco

crypto isakmp profile ISAKMP
   keyring KEYRING
   match identity address 123.0.0.2
   match identity address 123.0.0.3
   virtual-template 123

crypto ipsec transform-set TS esp-aes 256 esp-sha256-hmac
 mode tunnel

crypto ipsec profile IPSEC
 set transform-set TS
 set isakmp-profile ISAKMP

int lo1
 ip add 10.0.123.1 255.255.255.0

int virtual-template 123 type tunnel
 ip unnumbered loopback 1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile IPSEC
 no ip next-hop-self eigrp 123
 no ip split-horizon eigrp 123

router eigrp 123
 network 10.0.123.0 0.0.0.255
 network 192.168.0.1 0.0.0.0
```

**Spoke configuration**

```
crypto isakmp policy 10
 encr aes 256
 hash sha256
 authentication pre-share
 group 14
 lifetime 3600

crypto isakmp key cisco address 123.0.0.1
crypto isakmp key cisco address 123.0.0.3

crypto ipsec transform-set TS esp-aes 256 esp-sha256-hmac
 mode tunnel
```

```
crypto ipsec profile IPSEC
 set transform-set TS

interface tun0
 ip add 10.0.123.2 255.255.255.0
 tunnel source fa0/0
 tunnel destination 123.0.0.1
 tunnel mode ipsec ipv4
 ip mtu 1400
 tunnel protection ipsec profile IPSEC

router eigrp 123
 network 10.0.123.0 0.0.0.255
 network 192.168.0.2 0.0.0.0
```

# Static VTI

## Static Virtual Tunnel Interfaces (VTI)

- Default tunnel mode is GRE, optionally change to IPsec IPv4 / IPv6.
- The VPN will be initiated even if no traffic is generated.

```
crypto isakmp policy 10
 encr aes 256
 hash sha256
 authentication pre-share
 group 14
 lifetime 3600

crypto isakmp key cisco address 12.0.0.2

crypto ipsec transform-set TS esp-aes 256 esp-sha256-hmac
 mode tunnel

crypto ipsec profile IPSEC
 set transform-set TS

int tun0
 ip add 10.0.12.1 255.255.255.0
 tunnel source 12.0.0.1
 tunnel destination 12.0.0.2
 ip mtu 1400
 tunnel protection ipsec profile IPSEC
 qos pre-classify
```

```
router eigrp 12
  network 10.0.12.0 0.0.0.255
  network 192.168.0.1 0.0.0.0
```

# DMVPN

## Dynamic Multipoint VPN (DMVPN) Phases

| Phase | GRE-Mode | Dynamic Tunnels | Summarization / Default-Route |
|---|---|---|---|
| 1 | mGRE on hub, GRE on spokes | no | Allowed |
| 2 | mGRE on all | yes | Allowed, but lose dynamic tunnel functionality |
| 3 | mGRE on all | yes | Allowed |

**DMVPN Phase 1**

- Basically traditional Hub-and-Spoke topology without dynamic tunnels.
- Configure spokes with tunnel destination <hub nbma address> and tunnel mode gre on the tunnel interface.

**DMVPN Phase 3 Additions**

- The  ip nhrp redirect is configured on the hub and works similar to IP redirect. Informs spokes of the location of others.
- When a hub receives and forwards packet out of same interface it will sent a NHRP redirect message back to the source.
- The original packet from the source is not dropped but forwarded down to other spoke via RIB.
- The ip nhrp shortcut is configured on spokes and rewrites the CEF entry after getting the redirect message.

## Next Hop Resolution Protocol (NHRP)

- ARP-like protocol that dynamically maps a NBMA network.
- NHRP works similar to Frame-Relay DLCI but instead maps NMBA addresses to tunnel addresses.
- The hub can only communicate with spokes after spokes have registered to the hub.
- The hub is the Next Hop Server (NHS) and the spokes are the Next Hop Clients (NHCs).
- NHRP allows one NHC to dynamically discover another NHC within the network and build tunnels dynamically (resolution).

**NHRP Dynamic Flags**

| | |
|---|---|
| Authoritative | Obtained from NHS. |
| Implicit | Obtained from forwarded NHRP packet . |

| | |
|---|---|
| Negative | Could not be obtained. |
| Unique | Request packets are unique, disable if spoke has a dynamic outside IP address. |
| Registered | Obtained from NHRP registration request (Seen on hub).<br>The spoke has instructed the hub not to take a registration from another identical NBMA address. |
| Used | Set when data packets are process switched and mapping entry is in use, 60s timer. |
| Router | NHRP mapping entries for the remote router for access to network. |
| Local | Local network mapping. |

# BGP

## iBGP DMVPN Recommendations

- Configure Hub as Route-Reflector.
- Configure Spokes as Route-Reflector Clients.
- Use Peer-Groups for scalable config.

**P3 hub configuration**

```
int fa0/0
 ip add 123.0.0.1 255.255.255.0

int tun0
 tunnel source fa0/0
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip address 10.0.123.1 255.255.255.0
 ip nhrp redirect

router bgp 123
 bgp listen range 10.0.123.0/24 peer-group DMVPN
 bgp listen limit 2
 neighbor DMVPN peer-group
 neighbor DMVPN remote-as 123
 address-family ipv4
  neighbor DMVPN route-reflector-client
  neighbor DMVPN activate
```

**P3 spoke configuration**

```
int fa0/0
 ip add 123.0.0.2 255.255.255.0
```

```
int tun0
 tunnel source fa0/0
 tunnel mode gre multipoin
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast 123.0.0.1
 ip nhrp nhs 10.0.123.1 nbma 123.0.0.1
 ip address 10.0.123.2 255.255.255.0
 ip nhrp shortcut

router bgp 123
 neighbor 10.0.123.1 remote-as 123
 address-family ipv4
  neighbor 10.0.123.1 activate
```

# EIGRP

**EIGRP DMVPN Recommendations**

| Phase | Next-Hop-Self | Split-Horizon |
|---|---|---|
| 1 | Enabled | Disable when using specific routes<br>Enable when using default summary |
| 2 | Disable | Disable |
| 3 | Enabled | Disable when using specific routes<br>Enable when using default summary |

**EIGRP add-path support for DMVPN**

- Enables hubs to advertise up to four best paths to connected spokes.
- Disable next-hop-self for add-paths to operate.
- Can only be enabled named configuration.
- Should not be configured alongside variance.

**P3 hub configuration**

```
int fa0/0
 ip add 123.0.0.1 255.255.255.0

int tun0
 tunnel source fa0/0
```

```
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip address 10.0.123.1 255.255.255.0
 ip nhrp redirect

router eigrp DMVPN
 add ipv4 au 123
  network 10.0.123.0 0.0.0.255
 af-interface tun0
  summary-address 0.0.0.0/0
  no next-hop-self no-ecmp-mode
  no split-horizon
  add-paths 4
```

**P3 spoke configuration**

```
int fa0/0
 ip add 123.0.0.2 255.255.255.0

int tun0
 tunnel source fa0/0
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast 123.0.0.1
 ip nhrp nhs 10.0.123.1 nbma 123.0.0.1
 ip address 10.0.123.2 255.255.255.0
 ip nhrp shortcut

router eigrp DMVPN
 add ipv4 au 123
  network 10.0.123.0 0.0.0.255
```

# Misc

## DMVPN Encryption

- Mode tunnel adds 20 bytes overhead and is only used in a multi-tier DMVPN hub.
- One set of routers running cryptography and another set performing NHRP services.
- Mode transport is preferred for single-hub configurations.

```
crypto ipsec transform-set TS esp-aes 256 esp-sha256-hmac
 mode transport
```

## DMVPN QoS

- Apply a QoS policy on a DMVPN hub on a tunnel instance in the egress direction.
- Shape traffic for individual spokes (parent policy). Policy data flows going though the tunnel (child policy).
- Defined by NHRP group, each spoke can be managed individually using Per-Tunnel QoS.
- The spoke can belong to only one NHRP group per GRE tunnel interface.

```
policy-map R2_PM
 class class-default
  shape average 50 m
policy-map R3_POLICY
 class class-default
  shape average 25 m

int tun0
 ip nhrp map group R2 service-policy output R2_PM
 ip nhrp map group R3 service-policy output R3_PM

show policy-map multipoint
```

Spoke configuration

```
int tun0
 ip nhrp group R2
```

## DMVPN PIMv2

- PIM does not allow multicast traffic to leave the same interface it was received on.
- Override this behavior with ip pim nbma-mode configured on the tunnel interface.

# ODR

## DMVPN On-Demand Routing (ODR)

- Disable CDP on outside (physical) interfaces. Enable CDP on tunnel interfaces.
- ODR timers and CDP timers should be the same.
- Configure on hub only, spokes will receive a default route via ODR and will advertise their connected subnets.
- In P2 all traffic will still flow through the hub, ODR needs P3 in order to create dynamic tunnels.

**P3 hub configuration**

```
int fa0/0
 ip add 123.0.0.1 255.255.255.0

cdp timer 20
cdp holdtime 60
```

```
int tun0
 tunnel source fa0/0
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip address 10.0.123.1 255.255.255.0
 ip nhrp redirect
 cdp enable

router odr
 timers 20 60 60 90
```

**P3 spoke configuration**

```
int fa0/0
 ip add 123.0.0.2 255.255.255.0

cdp timer 20
cdp holdtime 60

int tun0
 tunnel source fa0/0
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast 123.0.0.1
 ip nhrp nhs 10.0.123.1 nbma 123.0.0.1
 ip address 10.0.123.2 255.255.255.0
 ip nhrp shortcut
 cdp enable
```

# OSPF

## OSPF DMVPN Recommendations

- It is vital to the workings of OSPF that the hub is the DR in broadcast and non-broadcast networks
- OSPF is not recommended for DMVPN because it is not possible to summarize within the same area
- The DMVPN topology is most likely going to be part of the backbone area
- PTMP will not create dynamic tunnels in P2. Broadcast is recommended for P2
- PTMP is recommended for P1 and P3

**P3 hub configuration**

```
int fa0/0
```

```
 ip add 123.0.0.1 255.255.255.0

int tun0
 tunnel source fa0/0
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip address 10.0.123.1 255.255.255.0
 ip ospf network point-to-multipoint
 ip ospf 123 area 0
 ip nhrp redirect
```

**P3 spoke configuration**

```
int fa0/0
 ip add 123.0.0.2 255.255.255.0

int tun0
 tunnel source fa0/0
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast 123.0.0.1
 ip address 10.0.123.2 255.255.255.0
 ip nhrp nhs 10.0.123.1 nbma 123.0.0.1
 ip ospf network point-to-multipoint
 ip ospf 123 area 0
 ip nhrp shortcut
```

## OSPF DMVPN Route Filter

- You can only filter routes from being added to the RIB, not LSAs from being received
- Basically filter every route on the spokes except for the default route

**Hub configuration**

```
router ospf 123
 default-information originate always
```

**Spoke configuration**

```
access-list 1 permit host 0.0.0.0
router ospf 123
 distribute-list 1 in
```

# RIP

**RIP DMVPN Recommendations**

| Phase | Split-Horizon |
|---|---|
| 1 | Disable when using specific routes<br>Enable when using default summary |
| 2 | Disable |
| 3 | Disable when using specific routes<br>Enable when using default summary |

**P3 hub configuration**

```
int fa0/0
 ip add 123.0.0.1 255.255.255.0

int tun0
 tunnel source fa0/0
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip address 10.0.123.1 255.255.255.0
 ip nhrp redirect

router rip
 version 2
 network 10.0.0.0
 no auto-summary
```

**P3 spoke configuration**

```
int fa0/0
 ip add 123.0.0.2 255.255.255.0

int tun0
 tunnel source fa0/0
 tunnel mode gre multipoint
 ip nhrp network-id 123
 ip mtu 1400
 ip nhrp map multicast 123.0.0.1
 ip nhrp nhs 10.0.123.1 nbma 123.0.0.1
 ip address 10.0.123.2 255.255.255.0
```

```
 ip nhrp shortcut

router rip
 version 2
 network 10.0.0.0
 no auto-summary
```

# IPv6

## IPv6 DMVPN over IPv4 NBMA

- IPv4 for the physical interface.
- IPv6 for the tunnel interface.
- Uses IPv4 NBMA address on nhs command.
- Uses IPv6 NHRP commands.

**P3 hub configuration**

```
int fa0/0
 ip add 10.0.123.1 255.255.255.0

interface tun0
 tunnel source 10.0.123.1
 tunnel mode gre multipoint ipv6
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 123
 ipv6 mtu 1400
 ipv6 address FE80::1 link-local
 ipv6 address 2001:10:0:123::1/64
 ipv6 nhrp redirect
```

**P3 spoke configuration**

```
int fa0/0
 ip add 10.0.123.2 255.255.255.0

interface tun0
 tunnel source 10.0.123.2
 tunnel mode gre multipoint
 ipv6 nhrp map multicast 123.0.0.1
 ipv6 nhrp network-id 123
 ipv6 nhrp nhs 2001:10:0:123::1 nbma 123.0.0.1
 ipv6 mtu 1400
 ipv6 address FE80::2 link-local
 ipv6 address 2001:10:0:123::2/64
 ipv6 nhrp shortcut
```

## IPv6 DMVPN over IPv6 NBMA

- IPv6 for the physical interface.
- IPv6 for the tunnel interface.
- Uses IPv6 NBMA address on nhs command.
- Uses IPv6 NHRP commands.

**P3 hub configuration**

```
int fa0/0
 ipv6 add 2001:123::1/64
 ipv6 address FE80::1 link-local

int tun0
 ipv6 mtu 1400
 tunnel source 2001:123::1/64
 tunnel mode gre multipoint ipv6
 ipv6 address FE80::1 link-local
 ipv6 address 2001:10:0:123::1/64
 ipv6 mtu 1400
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 123
 ipv6 nhrp redirect
```

**P3 spoke configuration**

```
int fa0/0
 ipv6 add 2001:123::2/64
 ipv6 address FE80::2 link-local

int tun0
 ipv6 mtu 1400
 tunnel source 2001:123::2/64
 tunnel mode gre multipoint ipv6
 ipv6 address FE80::2 link-local
 ipv6 address 2001:10:0:123::2/64
 ipv6 nhrp map multicast 2001:123:0:0::1
 ipv6 nhrp network-id 123
 ipv6 nhrp nhs 2001:10:0:123::1 nbma 2001:123::1
 ipv6 mtu 1400
 ipv6 nhrp shortcut
```

## IPv4 DMVPN over IPv6 NBMA

- IPv6 for the physical interface.
- IPv4 for the tunnel interface.
- Uses IPv6 NBMA address on nhs command.
- Uses IPv4 NHRP commands.

**P3 hub configuration**

```
int fa0/0
 ipv6 add 2001:123::1/64
 ipv6 address FE80::1 link-local

interface tun0
 ip mtu 1400
 tunnel source 2001:123::1
 tunnel mode gre multipoint ipv6
 ip nhrp map multicast dynamic
 ip nhrp network-id 123
 ip address 10.0.123.1 255.255.255.0
 ip nhrp redirect
```

**P3 spoke configuration**

```
int fa0/0
 ipv6 add 2001:123::2/64
 ipv6 address FE80::2 link-local

interface tun0
 ip mtu 1400
 tunnel source 2001:123::2
 tunnel mode gre multipoint ipv6
 ip nhrp map multicast dynamic
 ip nhrp network-id 123
 ip address 10.0.123.2 255.255.255.0
 ip nhrp nhs 10.0.123.1 nbma 2001:123::1
 ip nhrp shortcut
```

# EIGRP

## EIGRP Rules

- ASN (router eigrp xx) needs to match between neighbors
- Hello timers do not need to match between neighbors
- MTU size is used as a tie-breaker in path selection, it is not directly used in the metric formula
- supports MD5 authentication using key-chains (only named mode supports SHA authentication)
- Automatic summarization is on by default, manual summarization can only be configured on the links

## EIGRP Communication

- EIGRP uses Reliable Transport Protocol (RTP) for guaranteed, ordered delivery of EIGRP packets to all neighbors
- Supports multicast and unicast, and uses IP protocol 88
- Multicast address is 224.0.0.10 and FF02::A
- Only some EIGRP packets are sent reliably
    - For efficiency, reliability is provided only when necessary
    - EIGRP updates are sent to the multicast address and acknowledgements are replied via unicast

## EIGRP Timers

- The configured hold-time is communicated to the neighbor on the segment
    - This hold-time is included in the hello message
- The neighbor receives this and will expect a new hello from the router within this time
    - The default hello-interval is 5 seconds on (faster) LAN interfaces and 60 seconds for (slow) WAN interfaces
    - The default hold-timer is 3 times the hello-timer
    - These timers do not have to match

**Set the hello timer to 2 seconds and the hold timer to 6 seconds for AS1**

```
int gi0/0
  ip hello-interval eigrp 1 2
  ip hold-time eigrp 1 6
```

## EIGRP Protocol Messages

| Message | Purpose | Sent via | Contains |
|---|---|---|---|
| HELLO | Set up and maintain neighbors | Multicast to 224.0.0.10 / FF02::A<br>Unicast in case of static neighbors | Hello message |
| UPDATE | Contains topology information<br>EIGRP sends partial updates of information that is missing from the neighbors topology<br>Unicast is used for neighbor discovery<br>Multicast is for metric and prefix updates | Unicast RTP<br>Multicast RTP | Prefix<br>Prefix Length<br>Metric (bw, delay, reliability, load)<br>MTU, hopcount |
| ACK | Acknowledgement of the reception of an UPDATE<br>These messages are actually unicast HELLO messages | Unicast RTP | |

| | | | |
|---|---|---|---|
| QUERY | Sent when one or more destinations enter the active state. | Multicast RTP | |
| REPLY | Sent in response to a QUERY | Unicast RTP | |
| REQUEST | Requests information from neighbors | Unicast or Multicast | |

- Full updates are exchanged when neighbors first establish
- After network converges, updates are no longer exchanged (unless there is a change in the network or neighbors go down)
- If there is a change in the network, the prefixes that have changed are re-advertised to neighbors
    - Changes include changes in metric or status
    - Changing the bandwidth or delay on the interface will cause the router to re-advertise the route (if used in formula)
    - Changing the load or reliability will not cause to router to re-advertise the route

## Passive Interfaces

- Making an interface passive will stop EIGRP from sending multicast or unicast neighborship messages on that link
- The interface connect subnet will still be advertised in the EIGRP process
- Passive interfaces will only show up in the **show ip protocols** command
- Passive interfaces are not listed under **show ip eigrp interfaces**, this command only shows interfaces on which neighbors can form

## Command Output and Basic Config

- IOS is smart enough to know the difference between an inverted netmask and a subnet mask
- You can paste in your interface configuration directly into the network statement and it will automatically be converted to an inverted netmask
- This is still only an inverted netmask, so you cannot use the same matching techniques as with an ACL ( 0.1.7.255 for example)

**Below is a valid config**

```
int fa0/0
 ip add 10.0.12.2 255.255.255.252
 no shut

int lo1
 ip add 1.0.1.1 255.255.255.0
int lo2
 ip add 1.0.2.1 255.255.255.0
```

```
int lo3
 ip add 1.0.3.1 255.255.255.0
int lo4
 ip add 1.0.4.1 255.255.255.0

router eigrp 12
 network 10.0.12.2 255.255.255.252
 passive-interface default
 no passive-interface fa0/0

 network 1.0.1.1 255.255.255.0
 network 1.0.2.1 255.255.255.0
 network 1.0.3.1 255.255.255.0
 network 1.0.4.1 255.255.255.0
```

**IOS will translate this to**

```
router eigrp 12
 network 10.0.12.0 0.0.0.3
 passive-interface default
 no passive-interface FastEthernet0/0

 network 1.0.1.0 0.0.0.255
 network 1.0.2.0 0.0.0.255
 network 1.0.3.0 0.0.0.255
 network 1.0.4.0 0.0.0.255
```

```
R1#show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(12)
                              Xmit Queue   PeerQ        Mean   Pacing Time   Multicast
Interface             Peers   Un/Reliable  Un/Reliable  SRTT   Un/Reliable   Flow Timer
Fa0/0                   1         0/0        0/0          62       0/0           232
```

- The **show ip eigrp interfaces** command will only list the interfaces that are enabled for EIGRP and active (no passive-interfaces)
    - A helpful way to remember this is that all interfaces listed under this command will be able to form neighborships

```
R1#show ip eigrp interfaces detail
EIGRP-IPv4 Interfaces for AS(12)
                              Xmit Queue   PeerQ        Mean   Pacing Time   Multicast
Interface              Peers  Un/Reliable  Un/Reliable  SRTT   Un/Reliable   Flow Timer
Fa0/0                    1        0/0       0/0          34       0/1           96
  Hello-interval is 5, Hold-time is 15
  Split-horizon is enabled
  Next xmit serial <none>
  Packetized sent/expedited: 6/0
  Hello's sent/expedited: 183/3
  Un/reliable mcasts: 0/5  Un/reliable ucasts: 6/3
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 1  Out-of-sequence rcvd: 1
  Topology-ids on interface - 0
  Authentication mode is md5,  key-chain is "EIGRP_KEY"
```

- The **show ip eigrp interfaces detail** command will only list the interfaces that are enabled for EIGRP and active (no passive-interfaces)
    - This is the only command that will actually display the hello / hold timers and authentication settings

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 12"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(12)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 10.0.12.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 91 external 171
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    1.0.1.0/24
    1.0.2.0/24
    1.0.3.0/24
    1.0.4.0/24
    10.0.12.0/30
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 91 external 171
```

- This command and the routing table are the only commands that shows the (locally significant) eigrp distance
- This is the only command that shows the eigrp passive interfaces
- In this case the distance was altered to 91 internal and 171 external with the **distance eigrp 91 171** command
- This is the only command that displays the variance (unequal cost load balancing), used k-values, the maximum paths and the maximum hopcount
- This command and **show ip eigrp topology** display the router-id (RID)
    - Neighborships will still form if the RID is the same between two routers, the routers will not be installed in the router table however

```
R1# show ip eigrp events | i routerid
2    11:19:34.139 Ignored route, dup routerid int: 10.0.12.1
4    11:19:34.135 Ignored route, dup routerid int: 10.0.12.1
6    11:19:34.135 Ignored route, dup routerid int: 10.0.12.1
```

- The only real way to verify a route received from a duplicate router-id is with the **show ip eigrp events | i routerid** command
- This is a normal function of EIGRP to prevent routing loops and will show up in the events if a router receives its own routes back through neighbor updates

```
R1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(12)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   10.0.12.2               Fa0/0                    10 00:04:24   24   144  0  25
```

- Shows the current hold timer, round-trip time and query counter (a number here will indicate a stuck in active problem)
- Using the default hello-interval of 5 and a hold-timer of 15, a value under 10 seconds for the hold timer will probably mean that the neighborship is about to go down
- The **show ip eigrp neighbors detail** command will also show retransmits

```
R1#show ip eigrp traffic
EIGRP-IPv4 Traffic Statistics for AS(12)
  Hellos sent/received: 1255/1248
  Updates sent/received: 23/31
  Queries sent/received: 0/0
  Replies sent/received: 0/0
  Acks sent/received: 18/16
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 209
  PDM Process ID: 195
  Socket Queue: 0/10000/2/0 (current/max/highest/drops)
  Input Queue: 0/2000/2/0 (current/max/highest/drops)
```

- Displays the total amount of hello's sent and received, as well as the total amount of updates sent and received

```
R2#show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(12)/ID(10.0.12.2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 1.0.3.0/24, 1 successors, FD is 156160
        via 10.0.12.1 (156160/128256), FastEthernet0/0
P 1.0.1.0/24, 1 successors, FD is 156160
        via 10.0.12.1 (156160/128256), FastEthernet0/0
P 1.0.4.0/24, 1 successors, FD is 156160
        via 10.0.12.1 (156160/128256), FastEthernet0/0
P 1.0.2.0/24, 1 successors, FD is 156160
        via 10.0.12.1 (156160/128256), FastEthernet0/0
P 10.0.12.0/30, 1 successors, FD is 28160
        via Connected, FastEthernet0/0
```

- The first metric value is the feasible distance (FD) 156160 in this case, this is our distance to the route
- The second metric value is the reported or advertised distance (AD) 128256 in this case, this is the neighbors distance to the route

- Routes indicated with P are Passive, which means that EIGRP has converged on this route
- Routes indicated with A are Active, which means that EIGRP is actively looking for a next-hop of this route
    - This process will only start if the successor route has failed, and there is no feasible successor present
    - Going active on a route is sometimes referred to as DUAL (Diffusing Update Algorithm)

```
R2#show ip eigrp topology 1.0.1.0/24
EIGRP-IPv4 Topology Entry for AS(12)/ID(10.0.12.2) for 1.0.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 156160
  Descriptor Blocks:
  10.0.12.1 (FastEthernet0/0), from 10.0.12.1, Send flag is 0x0
      Composite metric is (156160/128256), route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 5100 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 10.0.12.1
```

- On FastEthernet the reliability is usually 255 (unless there are errors on the link), the load is 1 and the MTU is 1500

```
R1#show ip route 4.0.1.0 255.255.255.0
Routing entry for 4.0.1.0/24
  Known via "eigrp 1", distance 90, metric 158720, type internal
  Redistributing via eigrp 1
  Last update from 10.0.13.3 on FastEthernet0/1, 00:05:57 ago
  Routing Descriptor Blocks:
    10.0.13.3, from 10.0.13.3, 00:05:57 ago, via FastEthernet0/1
      Route metric is 161536, traffic share count is 59
      Total delay is 5200 microseconds, minimum bandwidth is 90000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
  * 10.0.12.2, from 10.0.12.2, 00:05:57 ago, via FastEthernet0/0
      Route metric is 158720, traffic share count is 60
      Total delay is 5200 microseconds, minimum bandwidth is 100000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

- Like the topology table, the standard routing table also shows the eigrp metric values, as well as the traffic share count (UCMP in this case)

# Filtering

## Extended Access-List

**Deny even from R2 and odd from R3**

```
ip access-list extended 100
 deny ip host 10.0.12.2 4.0.0.0 0.0.0.254
 deny ip host 10.0.13.3 4.0.0.1 0.0.0.254
 permit ip any any

router eigrp 1
 distribute-list 100 in
```

## Prefix-List

**Deny all prefixes from R2**

```
ip prefix-list DENY_R2 deny 10.0.12.2/32
ip prefix-list DENY_R2 permit 0.0.0.0/0 le 32
ip prefix-list PREFIXES permit 0.0.0.0/0 le 32

router eigrp 1
```

```
distribute-list prefix PREFIXES gateway DENY_R2 in
```

**Only accept prefixes from R3**

```
ip prefix-list R3 permit 10.0.13.3/32

router eigrp 1
 distribute-list gateway R3 in
```

**Deny specific prefixes from R2**

```
ip prefix-list R2 permit 10.0.12.2/32
ip prefix-list NETWORKS deny 4.0.0.1/32
ip prefix-list NETWORKS deny 4.0.0.2/32
ip prefix-list NETWORKS permit 0.0.0.0/0 le 32

router eigrp 1
 distribute-list prefix NETWORKS gateway R2 in fa0/0
```

# Metric

## EIGRP Composite Metric (Weight Calculation)

- EIGRP uses metric weights along with a set of vector metrics to compute the composite metric for local RIB and route selections.
- Type of service (first K value) must always be zero.
- The formula is [K1*bandwidth + (K2*bandwidth)/(256 - load) + K3*delay] * [K5/(reliability + K4)].

256*[(10^7/slowest bandwidth in kbps) + all link delays in tens microseconds]

| Interface Type | Bandwidth | Delay in µsec | Delay in tens of µsec |
|---|---|---|---|
| Loopback | 8,000,000 | 5000 | 500 |
| Ethernet | 10,000 | 1000 | 100 |
| FastEthernet | 100,000 | 100 | 10 |
| GigabitEthernet | 1,000,000 | 10 | 1 |
| Serial | 1544 | 20000 | 2000 |

**Example Loopback route via two FastEthernet hops:**
Slowest bandwidth is 100000 kbps
Total delay is (100 x2) + 500 = 520

Formula = 256*[(10^7/100000) + 520)] = 256*(100+520) = 158,720

```
R1#sh ip eigrp topology 4.0.1.0/24
EIGRP-IPv4 Topology Entry for AS(1)/ID(10.0.13.1) for 4.0.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 158720
  Descriptor Blocks:
  10.0.12.2 (FastEthernet0/0), from 10.0.12.2, Send flag is 0x0
      Composite metric is (158720/156160), route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 5200 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
        Originating router is 4.0.4.1
```

```
router eigrp
 metric weights 0 1 0 1 0 0
```

## EIGRP Wide Metrics

- The EIGRP Wide Metric feature supports 64-bit metric calculations and Routing Information Base (RIB) scaling.
- The lowest delay that can be configured for an interface is 10 microseconds with 32-bit metrics (normal).
- Use on interfaces that are faster than 1Gbps.
- The new metrics no longer fit in the output of the RIB, because this is limited to 32bit numbers.
- The topology table will show the correct metric numbers which is divided by the value specified in the rib-scaling.
- The formula is [K1*bandwidth + (K2*bandwidth)/(256 - load) + K3*delay +K6*Ext Attr] * [K5/(reliability + K4)]
- K6 is an additional K value for future use. Other K values remain the same with K1 and K3 set to 1, and K2,K4,K5 set to 0.

```
router eigrp EIGRP
 address-family ipv4 unicast autonomous-system 1
  metric rib-scale 128
```

The 64-bit metric calculations work only in EIGRP named mode configurations. EIGRP classic mode uses 32-bit metric calculations.

- EIGRP named mode automatically uses wide metrics when speaking to another EIGRP named mode process.

## EIGRP Offset Lists

- The offset list increases the existing metric by a specified amount. It is not possible to decrease a metric.
- The offset list modifies the delay value, not the bandwidth value. Meaning that it is included in the cumulative delay.

- Using **offset-list 0** will apply the offset to all networks. Using an empty access list has the same effect.

An offset list will only influence the calculated metric, and thus the composite metric

- This composite metric includes both the feasible distance AND the reported distance, making it very hard to use offset-list to create a specific UCMP load scenario
- The composite metric is only used for local calculation and is not communicated to neighbors
- However, routing paths through the router between neighbors will still add the offset value to the metric.
- This is because the offset list changes the delay value which is cumulative in the total metric for the route

## EIGRP Unequal Cost Multi-Path Load-Sharing

- Only feasible successors are a candidate for load balancing.
- The **traffic-share min across-interfaces** command only uses the primary path in case of a UCMP configuration using variance.
- This will stop the route from using UCMP. The point of this configuration is to speed up convergence.
- The **variance** command is just a multiplier of how much 'worse' the other route can be, if you don't have any feasible successors a variance of any value up to 128 will not change anything
- When calculating the eigrp traffic share count, divide numbers to figure out metric.
- A preferred traffic share count of two routes, one 10 the other 3 -> 10/3 = 3.33 * metric value.

```
router eigrp 1
 variance 5
 traffic-share min across-interfaces
```

## Default Metric & Redistribution

- Only connected and static routes can be redistributed without a default metric. This metric is set to 0.
- IPv6 networks add the **include-connected** keyword to include connected networks as well.
- In IPv4 connected networks are included in the redistribution by default.

## Metric Maximum-Hops

- The maximum hops is a greater than statement. Default is 100 hops.
- If 10 is entered for example, the prefixes 10 hops away are still valid. The prefixes 11 hops away are denied.

```
router eigrp 1
 metric maximum-hops 10
```

# Misc

## EIGRP Neighbor Statement

- Limit neighbor ship with specific neighbors by configuring neighbor statements (on the same segment)
- This will ensure that EIGRP hellos are sent unicast to the specified neighbor and not multicast to all neighbors to 224.0.0.10
- Configuring the neighbor statement will essentially stop the router from sending multicast messages to 224.0.0.10, meaning that all neighborships are dropped on that interface unless they are specifically configured in the neighbor statement
- You need to specify an interface with a neighbor ip-address
- The neighbor statement will only work on interfaces that are active for EIGRP (network statement)

```
router eigrp 1
 neighbor 10.0.12.2 fa0/0
```

```
R1#show ip eigrp neighbors detail
EIGRP-IPv4 Neighbors for AS(1)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   10.0.12.2               Fa0/0                    11 00:00:08   28   168  0  3
   Static neighbor
   Version 10.0/2.0, Retrans: 0, Retries: 0
   Topology-ids from peer - 0
```

- Will show up as 'static neighbor' using the **show ip eigrp interfaces detail** command

**Block specific EIGRP neighbors**

- One option is neighbor statements
- Alternative is to use ACL to block specific link-local addresses (IPv6) for EIGRP traffic (protocol 88)

```
ipv6 router eigrp 1
 neighbor FE80::2 fa0/0
 neighbor FE80::3 fa0/0
```

**Or**

```
ipv6 access-list EIGRP_BLOCK
 deny 88 fe80::2/128  any
 deny 88 fe80::3/128  any
 permit ipv6 any any

int fa0/0
 ipv6 traffic-filter EIGRP_BLOCK in
```

## EIGRP Redistribution

- Normally, static routes (or other protocols) can only be advertised into EIGRP with the redistribute command

```
ip route 0.0.0.0 0.0.0.0 100.0.0.1
router eigrp 1
 redistribute static 1000000 1 255 1 1500
```

- The exception is static routes that ONLY point to an outgoing interface and not a next-hop IP address
    - These routes will be advertised into the EIGRP process if the network statement matches the static route subnet

```
ip route 192.168.0.0 255.255.255.0 GigabitEthernet 0/0
router eigrp 1
 network 192.168.0.0 0.0.0.255
```

**Or**

```
ip route 0.0.0.0 0.0.0.0 GigabitEthernet 0/0
router eigrp 1
 network 0.0.0.0
```

```
R2#show ip eigrp topology 0.0.0.0/0
EIGRP-IPv4 Topology Entry for AS(1)/ID(20.0.0.2) for 0.0.0.0/0
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2816
  Descriptor Blocks:
  0.0.0.0, from Rstatic, Send flag is 0x0
      Composite metric is (2816/0), route is Internal
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 10 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
        Originating router is 20.0.0.2
      Exterior flag is set
```

- This is only true for static routes that have no next-hop IP address and only an outgoing interface

## EIGRP Router-ID

- Duplicate router-IDs (RID) do not show up in the logging or debug. Use the **show ip eigrp events | i dup** command.
- EIGRP uses the concept of the RID as a loop-prevention mechanism to filter out a routers own routes.
- In the event of a duplicate RID the neighbors routes will not be installed.

## EIGRP Authentication

- It is possible to configure multiple keys. Only one authentication packet is sent, regardless of how many keys exist

- The software examines the key numbers in the order from lowest to highest, and uses the first valid key that it encounters
    - The **key chain name** is locally significant, only the **key-string** and the **key ID** need to match between neighbors
- HMAC-SHA-256 authentication is only available in named mode. Does not support key-chains
- Named mode ignores all authentication (and other) commands configured on the local interface using the old method

```
key chain EIGRP_KEY
 key 10
  key-string cisco

int fa0/0
 ip authentication key-chain eigrp 1 EIGRP_KEY
 ip authentication mode eigrp 1 md5
```

**HMAC-SHA-256 using named configuration**

```
router eigrp EIGRP
 address-family ipv4 autonomous-system 1
  af-interface fa0/0
   authentication mode hmac-sha-256 cisco
```

## Dampening Change & Interval

- Dampening controls the update of metric changes of routes advertised by neighbors.
- Dampening compares the old metric for the route with the new metric.
- Using dampening-change, if this new metric is within the percentage threshold, the update will be ignored.
- Using dampening-interval, if this new metric is within the configured time interval, the update will be ignored.
- Dampening is disabled by default.

```
int fa0/0
 ip dampening-change eigrp 1 percent 75
 ip dampening-interval eigrp 1 seconds 60
```

## Next-Hop-Self

- Disable next-hop-self on the redistributing router when using 3rd party next hop.
- This is used when OSPF and EIGRP coexist on the same segment, and one router is used for redistributing both protocols.
- Normally all traffic would go through the redistributing router, with 3rd party next hop the neighbors can communicate directly.
- Requirement is to disable next-hop-self on the interface of the redistributing router towards the shared segment.
- Other situations where next-hop-self might be disabled is in DMVPN solutions.

```
int fa0/0
 description SHARED_OSPF_EIGRP_SEGMENT
 ip add 10.0.123.1 255.255.255.0
 no ip next-hop-self eigrp 1

router eigrp 1
 network 10.0.123.0 0.0.0.255
 redistribute ospf 1

router ospf 1
 network 10.0.123.0 0.0.0.255 area 0
 redistribute eigrp 1 subnets
```

## EIGRP WAN Bandwidth Control (Bandwidth Percentage)

- By default, EIGRP packets are allowed to consume a maximum of 50 percent of the link bandwidth
- The bandwidth is the configured bandwidth with the **bandwidth** statement, not the original interface bandwidth
    - Bandwidth is specific to each sub-interface, bandwidth configured on the physical interface is not inherit by sub-interfaces

- EIGRP achieves bandwidth control by queuing the messages in memory, a form of QoS shaping specific to EIGRP messages
- Note that values greater than 100 percent may be configured
    - This configuration option may be useful if the bandwidth is set artificially low for other reasons.

```
int se0/0
 bandwidth 512
 ip bandwidth-percent eigrp <as> <percentage>
 ip bandwidth-percent eigrp 1 75
```

- The default bandwidth on serial interfaces is 1544 kbps
- When using frame-relay, each PVC configured on the (sub)interface will divide the bandwidth by that number
    - For example, if three PVCs are configured on a multipoint interface the (configured interface) bandwidth is divided by three
    - The percentage of the bandwidth that EIGRP packets are allowed to use is 50% of that by default

**Three PVCs with 50% of bandwidth:**

```
int se0/0
 no shutdown
 encapsulation frame-relay
 bandwidth 1544
```

```
int se0/0.123 multipoint
 bandwidth 1544
 frame-relay interface-dlci 102
 frame-relay interface-dlci 103
 frame-relay interface-dlci 104

 no ip split-horizon eigrp 1234
 ip bandwidth-percent eigrp 1234 50
```

1544 / 3 = ~515 kbps / 2 = ~257 kbps

### Administrative Distance

- Changes to the administrative distance of all internal and external routes is limited to the router itself.
- Can be limited to a specific neighbor, only one specific distance command can be entered per neighbor.

```
ip access-list standard NETWORK
 permit host 172.16.0.0

router eigrp 1
 distance eigrp 90 170
 distance 85 10.0.12.2 0.0.0.0
 distance 75 10.0.13.3 0.0.0.0 NETWORK
```

### EIGRP Loop-Free Alternate Fast Reroute (FRR)

- Uses repair paths or backup routes and installs these paths in the RIB.
- FRR picks the best feasible successor and places it in the FIB as a backup route.

```
router eigrp EIGRP
 address-family ipv4 autonomous-system 1
  topology base
   fast-reroute per-prefix all
```

# Route-Tags

### Route Tag Enhancements

- The route tag enhancements allow the route tag to be formatted as a dotted decimal tag.
- These can me matched either directly (in the traditional route tag method in route-map) or via a route-tag list.
- EIGRP named mode provides the option of assigning a **default-route-tag** to all routes sourced by the router.

```
route-tag notation dotted-decimal
ip access-list standard Lo1
 permit host 4.0.0.1
ip access-list standard Lo2
 permit host 4.0.0.2

ip prefix-list Lo3 permit 4.0.0.3/32

route-map LOOPBACKS permit 10
 match ip address Lo1
 set tag 44.44.1.1
route-map LOOPBACKS permit 20
 match ip address Lo2
 set tag 44.44.2.1
route-map LOOPBACKS permit 30
 match ip address prefix-list Lo3
 set tag 44.44.3.1
route-map LOOPBACKS permit 40
 match interface loopback 4
 set tag 44.44.4.1

router eigrp 1
 redistribute connected route-map LOOPBACKS
```

**Route-tag Even Filtering**

```
route-tag notation dotted-decimal
route-tag list LOOPBACKS permit 44.44.0.0 0.0.254.255

route-map LOOPBACKS permit 10
 match tag list LOOPBACKS
 set metric 50000 581 255 1 1500
route-map LOOPBACKS permit 20

router eigrp 1
 distribute-list route-map LOOPBACKS in fa0/0
```

**Default Route-Tags**

```
router eigrp EIGRP
 address-family ipv4 unicast autonomous-system 1
  eigrp default-route-tag 192.168.0.1
```

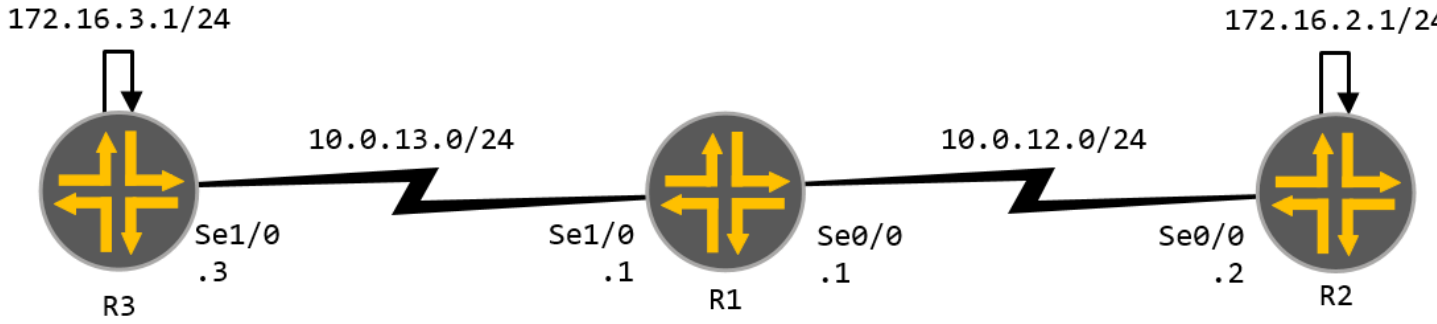
# Summarization

## EIGRP Route Summarization

- Summary routes are always internal, even if external routes are summarized.
- Adding a summary route will reestablishes the EIGRP neighborship with the neighbor on that specific link (not all links)
- The summary route will use a metric equal to the metric of the lowest metric of the routes included in the summary

## EIGRP Auto Summary

- Is only applicable to class A, B or C networks (172.16.0.0/10, 10.0.0.0/8, 192.168.0.0/16)
- When enabled, if the router has multiple interfaces in a classful network, it will attempt to summarize the routes towards neighbors
- Can cause issues with discontiguous networks, in other words networks that consist of many classless routes coming from multiple destinations




**R2**

```
int se0/0
 ip add 10.0.12.1 255.255.255.0
 no shut
int lo0
 ip add 172.16.2.1 255.255.255.0

router eigrp 123
 network 10.0.12.0 0.0.0.255
 network 172.16.2.0 0.0.0.255
 auto-summary
```

- With **auto-summary** R1 will receive a 172.16.0.0/10 route from both R3 and R2 instead of the separate 172.16.3.0/24 and 172.16.2.0/24 classless networks

## EIGRP Default Route Generation

- A default route can be advertised into EIGRP using three methods
  - Method 1. Default static route towards next-hop IP address (redistribute)
  - Method 2. Default static route towards outgoing interface (redistribute or network command)
  - Method 3. Default network statement (**ip default-network** command)

**Method 1**

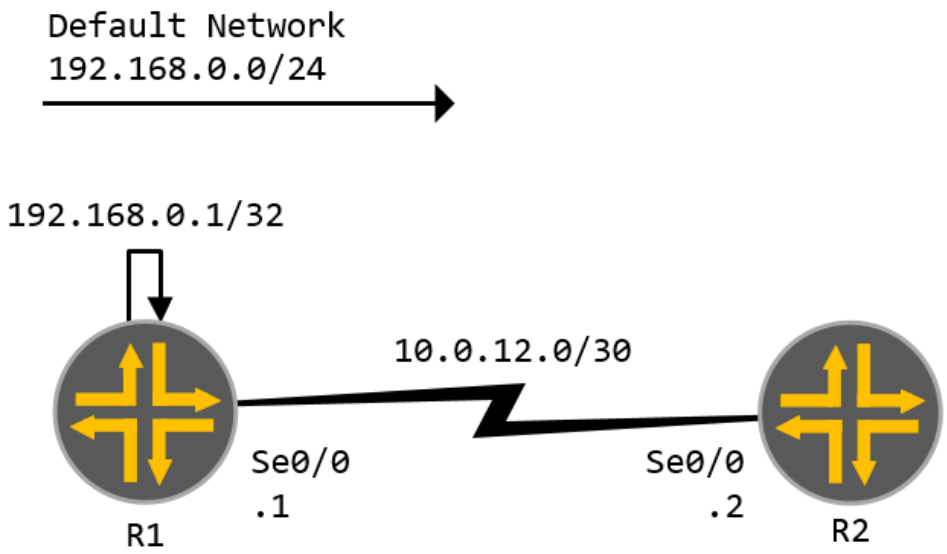
```
ip route 0.0.0.0 0.0.0.0 100.0.0.1
router eigrp 1
 redistribute static metric 1000000 1 255 1 1500
```

**Method 2**

```
ip route 0.0.0.0 0.0.0.0 GigabitEthernet 0/0
router eigrp 1
 network 0.0.0.0
```

**Method3**

- Marks a classful network as a route that can be used as a default route
- This is called a candidate default route and is indicated with a * in **show ip route**
- The classful network needs to be advertised into EIGRP using the **network** statement or via redistribution
- Preferably use a loopback and advertise that route as the default network to the rest of the EIGRP domain
- The behavior of this command seems to be different depending on the version of IOS
    - On 15.* the gateway of last resort does not seem to be added to neighbor routing tables
    - On 12.* the gateway of last resort is added to neighbor routing tables




```
int se0/0
 ip add 10.0.12.1 255.255.255.252
 no shut
int lo0
 ip add 192.168.0.1 255.255.255.0

ip default-network 192.168.0.0
```

```
router eigrp 12
 network 10.0.12.0 0.0.0.255
 network 192.168.0.0 0.0.0.255
```

- The order of configuration is important, first configure the **ip default-network** command, afterwards advertise the route into EIGRP
- Make sure to verify that the exterior flag is set on the route in the EIGRP topology

```
R1#show ip eigrp topology 192.168.0.0/24
IP-EIGRP (AS 1): Topology entry for 192.168.0.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 128256
  Routing Descriptor Blocks:
  0.0.0.0 (Loopback0), from Connected, Send flag is 0x0
      Composite metric is (128256/0), Route is Internal
      Vector metric:
        Minimum bandwidth is 10000000 Kbit
        Total delay is 5000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1514
        Hop count is 0
      Exterior flag is set
```

!----------!!----------!!----------!!----------!

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.12.0/30 is directly connected, Serial0/0
D       10.0.0.0/8 is a summary, 00:02:56, Null0
C*   192.168.0.0/24 is directly connected, Loopback0
```

!----------!!----------!!----------!!----------!

```
R1#show ip route 192.168.0.0 255.255.255.0
Routing entry for 192.168.0.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface), candidate default path
  Redistributing via eigrp 1
  Routing Descriptor Blocks:
  * directly connected, via Loopback0
      Route metric is 0, traffic share count is 1
```

!----------!!----------!!----------!!----------!

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.0.12.1 to network 192.168.0.0

     10.0.0.0/30 is subnetted, 1 subnets
C       10.0.12.0 is directly connected, Serial0/0
D*   192.168.0.0/24 [90/2297856] via 10.0.12.1, 00:00:47, Serial0/0
```

## Discard Route & Leak-Map

- The summary route has an AD of 5 and is called the discard route which points to Null0 on the summarizing router
- The neighbor will receive an internal route with an AD of 90
- Disable the discard route by specifying a **summary-metric** distance of 255
- Configure a leak-map to allow components of the summary to advertised alongside the summary
- Stop IP ICMP unreachables based on discard route with **no ip unreachables** configured on null0 interface

```
int fa0/0
 ip summary-address eigrp 1 0.0.0.0/0
```

**Disable discard route**

```
router eigrp 1
 summary-metric 0.0.0.0/0 distance 255
```

**Disable ICMP unreachables (silently drop traffic)**

```
int null0
 no ip unreachables
```

**Allow the 10.0.12.0/24 network to be advertised alongside the summary**

```
ip prefix-list NETWORK permit 10.0.12.0/24
route-map LEAK_MAP permit 10
 match ip address prefix-list NETWORK

int fa0/0
 ip summary-address eigrp 1 10.0.0.0/16 leak-map LEAK_MAP
```

# Stub

## EIGRP Stuck in Active

- EIGRP query messages live for 3 minutes by default. Can be modified with the **timers active-time** command
- Queries are sent (A in topology table) if a successor route is lost, the router will seek for a new next-hop of the route
- Queries are not sent if a feasible successor is present in the topology for the route
- A router will wait for a REPLY for all neighbors before failing over to a different path.
- If a router does not receive a reply to a QUERY within the configured active time (3 min) it will consider the route Stuck in Active (SIA)
    - Earlier versions of IOS (<12.2) just terminated the neighborship at the end of the **active-time**
    - More recent versions send a SIA-QUERY at the half-life of the **active-time** (90 sec) to verify if the neighbor is also waiting for replies
    - If the router receives a SIA-REPLY back it will continue waiting and not terminate the neighborship, if the neighbor does not reply the neighborship is terminated at the end of the active time

- SIA can be fixed with either STUB routers, or summary routes (or a combination of both)
    - Using summaries. If a router receives an EIGRP QUERY for a route that is part of a locally configured summary, the router immediately sends a REPLY and does not flood the QUERY to its own neighbors

## EIGRP Stub

- Fix for Stuck in Active by limiting queries to routers that are configured as stub
- Stub routers are limited in what they advertise to the rest of the network
    - Stub routers do NOT advertise any EIGRP-learned routes from one neighbor to other EIGRP neighbors
    - Stub routers ONLY advertise connected routes, summary routes (if specified) that they themselves create, and redistributed routes (if specified) that they themselves redistribute
- Neighbors are informed that queries for specific networks do not have to be sent to the stub router
- Configuring a router as stub will reestablish the adjacency
- A leak map can allow routes that would normally have been suppressed (only applies to summary routes)
- These routes need to be part of the stub option, you cannot for example allow specific redistributed routes while only using **stub connected**

Stub Options

| **Keyword** | **Description** | **Advertises** | **Receives** | **Forwards** |
|---|---|---|---|---|
| RECEIVE-ONLY | Router only receives and advertises nothing | - | All | - |
| CONNECTED | Router only advertises connected subnets | connected | All | - |
| STATIC | Router only advertises static routes | static | All | - |
| SUMMARY | Router only advertises summary routes | summary | All | - |
| REDISTRIBUTE | Router only advertises redistributed routes | external | All | - |

- The default when you configure **eigrp stub**, is **eigrp stub connected summary**
- Stub routers will show up as stub (with the configured keyword) using the **show ip eigrp neighbors detail** command

```
R1#show ip eigrp neighbors detail
EIGRP-IPv4 Neighbors for AS(1)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   10.0.13.3               Gi1/0                    11 00:01:54  692  4152  0  39
   Time since Restart 00:00:37
   Version 10.0/2.0, Retrans: 1, Retries: 0
   Topology-ids from peer - 0
   Stub Peer Advertising (SUMMARY ) Routes
   Suppressing queries
```

**Allow prefixes from 172.16.1.0/24 alongside the connected and summary routes**

```
int fa0/0
 ip summary-address eigrp 1 172.16.0.0/16

ip prefix-list STUB_PREFIX permit 172.16.1.0/24
route-map STUB_LEAK_MAP permit 10
 match ip address prefix STUB_PREFIX

router eigrp 1
 eigrp stub connected summary leak-map STUB_LEAK_MAP
```

# Frame-Relay

## DCE & DTE

- L1 DTE = Data Termination Equipment
- L1 DCE = Data Communication Equipment
- Frame-Relay (FR) switch is called the DCE, endpoints are called DTE
    - This is unrelated to L1 DCE/DTE which sets the clocking rate of the serial interface
    - A Frame-Relay DTE device can be the DCE for the serial interface and vice versa

## L1 and Frame-Relay Terminology

- FR DCE responds to LMI inquiries by sending LMI status, never sends LMI inquiries
- FR DTE sends LMI inquiries, never sends LMI status
- DLCIs are only locally significant

## Local Management Interface (LMI)

- Signaling method to exchange IP information, PVCs (Permanent Virtual Circuits) and keepalive messages

- Basically LMIs take care of the hello and update messages between a DTE device and a FR switch (DCE)

**LMI Status Messages**

| Status | Description | Reason for problem |
|---|---|---|
| STATIC | LMI is disabled on the interface | DLCI is configured on the local device, but FR switch does not have a route |
| INACTIVE | LMI is enabled on the interface and configured correctly<br>FR switch is configured with the frame route<br>Remote device does not have the correct DLCI configured | PVC is configured correctly on the local switch, but there is a problem on the remote end of the PVC (other device) |
| DELETED | Misconfiguration on local device<br>FR switch is configured with the correct DLCI | FR switch has deleted the frame route or local device is misconfigured (usually goes along with STATIC status) |
| ACTIVE | The PVC is operational and can transmit packets | - |

```
R1#show frame-relay map
Serial4/0.2 (up): point-to-point dlci, dlci 123(0x7B,0x1CB0), broadcast
          status defined, active
Serial4/0.3 (up): point-to-point dlci, dlci 133(0x85,0x2050), broadcast
          status defined, active
Serial4/0.4 (down): ip 10.0.4.3 dlci 143(0x8F,0x20F0), static,
              CISCO, status deleted
```

## Inverse ARP

- Because serial interfaces and FR are NBMA networks, there is no ARP flooding to discover neighbor addresses
- Instead FR relies on inverse ARP to allow clients to notify neighbors of their presence and reply to requests (dynamic mappings)
- Disabling inverse ARP will disable these notifications, disabling inverse ARP will require static mappings
    - Inverse ARP has to be disabled on all routers to disable automatic learning of mappings, if you forget a single device, inverse ARP will still work
    - Inverse-ARP does not allow for self-ping, static mappings DO allow for self-ping

**Disable inverse arp**

```
int s1/0
 encapsulation frame-relay
 no frame-relay inverse-arp

clear frame inarp
```

## Frame-Relay Static Mappings

- If dynamic mappings have been disabled, static ones need to be created on the clients
- The broadcast keyword allows broadcast packets over NBMA
    - This is not real broadcast traffic (pseudo-broadcast), instead broadcast packets are simply sent as unicast but repeated for each and every client
- Only configure the **broadcast** keyword on one **frame-relay map** statement, preferably the statement that configures the router to ping itself
    - If you configure it on multiple lines, the router will respond with multiple replies to a (pseudo)broadcast message

```
int se1/0
 ip address 10.0.13.1 255.255.255.0
 frame-relay map ip 10.0.13.1 103 broadcast
 frame-relay map ip 10.0.13.3 103

int se1/0
 ip address 10.0.13.3 255.255.255.0
 frame-relay map ip 10.0.13.1 301
 frame-relay map ip 10.0.13.3 301 broadcast

show frame pvc
show frame map
```

## Frame-Relay Switch

- Set encapsulation to frame-relay
- Configure DCE on interfaces with the **frame-relay intf-type dce** command (default type is DTE)
- Create FR PVCs using either frame routes (FR switching framework) or connections (L2VPN framework)

**Create FR PVCs using frame routes on FR Switch**

```
int se1/0
 encapsulation frame-relay
 frame-relay intf-type dce
 frame route 103 interface se1/1 301

int se1/1
 encapsulation frame-relay
 frame-relay intf-type dce
 frame route 301 interface se1/0 103

show frame route
show frame pvc
```

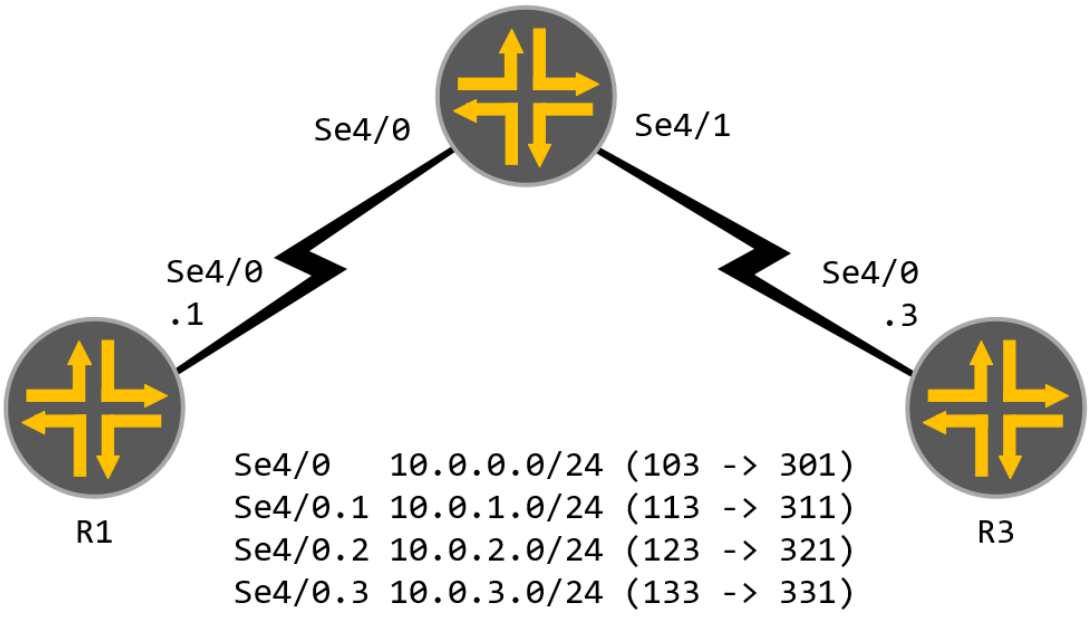
**Create FR PVCs using connections on FR Switch**

```
int se1/0
 encapsulation frame-relay
 frame-relay intf-type dce

int se1/1
 encapsulation frame-relay
 frame-relay intf-type dce

connect R1-R3 se1/0 103 se1/1 301
show connection all
```

# Point-to-Point

## Frame-Relay with Point-to-Point (P2P) Sub-Interfaces

- The DLCI needs to be linked to the sub-interface
- Inverse ARP configuration is not inherited by sub-interfaces (default is on)
  - The **no inverse-arp** command on the physical interface is not inherited by the sub-interface
- P2P sub-interfaces only have two endpoints, for this reason the pseudo-broadcast statement is not required because there is only one destination
  - For the same reason, the only thing that is significant is the local DLCI configured with the **frame-relay interface-dlci** command
  - There is no need to map IPs to DLCIs or to include the **broadcast** keyword
  - Limited to one DLCI per sub-interface




**R1**

```
int se4/0
```

```
 no shutdown
 encapsulation frame-relay
 ip add 10.0.0.3 255.255.255.0

 no frame inverse-arp
 frame-relay map ip 10.0.0.1 103  broadcast
 frame-relay map ip 10.0.0.3 103

int se4/0.1 point-to-point
 ip add 10.0.1.1 255.255.255.0
 no frame inverse-arp
 frame-relay interface-dlci 113

int se4/0.2 point-to-point
 ip add 10.0.2.1 255.255.255.0
 no frame inverse-arp
 frame-relay interface-dlci 123

int se4/0.3 point-to-point
 ip add 10.0.3.1 255.255.255.0
 no frame inverse-arp
 frame-relay interface-dlci 133
```

**Frame Relay Switch (R2)**

```
frame-relay switching
int se4/0
 encapsulation frame-relay
 frame-relay intf-type dce
 no shutdown
 frame-relay route 103 interface s4/1 301
 frame-relay route 113 interface s4/1 311
 frame-relay route 123 interface s4/1 321
 frame-relay route 133 interface s4/1 331

int se4/1
 encapsulation frame-relay
 frame-relay intf-type dce
 no shutdown

 frame-relay route 301 interface s4/0 103
 frame-relay route 311 interface s4/0 113
 frame-relay route 321 interface s4/0 123
 frame-relay route 331 interface s4/0 133
```

**R3**

```
int se4/0
```

```
 no shutdown
 encapsulation frame-relay
 ip add 10.0.0.3 255.255.255.0

 no frame inverse-arp
 frame-relay map ip 10.0.0.1 301
 frame-relay map ip 10.0.0.3 301 broadcast

int se4/0.1 point-to-point
 ip add 10.0.1.3 255.255.255.0
 no frame inverse-arp
 frame-relay interface-dlci 311

int se4/0.2 point-to-point
 ip add 10.0.2.3 255.255.255.0
 no frame inverse-arp
 frame-relay interface-dlci 321

int se4/0.3 point-to-point
 ip add 10.0.3.3 255.255.255.0
 no frame inverse-arp
 frame-relay interface-dlci 331
```

## Verify Output

```
R2#show frame-relay route
Input Intf      Input Dlci      Output Intf     Output Dlci     Status
Serial4/0       103             Serial4/1       301             active
Serial4/0       113             Serial4/1       311             active
Serial4/0       123             Serial4/1       321             active
Serial4/0       133             Serial4/1       331             active
Serial4/1       301             Serial4/0       103             active
Serial4/1       311             Serial4/0       113             active
Serial4/1       321             Serial4/0       123             active
Serial4/1       331             Serial4/0       133             active

!----------------------!!----------------------!!----------------------!!----------------------!
```

```
R2#show frame-relay pvc 103

PVC Statistics for interface Serial4/0 (Frame Relay DCE)

DLCI = 103, DLCI USAGE = SWITCHED, PVC STATUS = ACTIVE, INTERFACE = Serial4/0

  input pkts 5             output pkts 5            in bytes 520
  out bytes 520            dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                 out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
  switched pkts 5
  Detailed packet drop counters:
  no out intf 0            out intf down 0          no out PVC 0
  in PVC down 0            out PVC down 0           pkt too big 0
  shaping Q full 0         pkt above DE 0           policing drop 0
  connected to interface Serial4/1 301
  pvc create time 00:03:30, last time pvc status changed 00:02:20
```

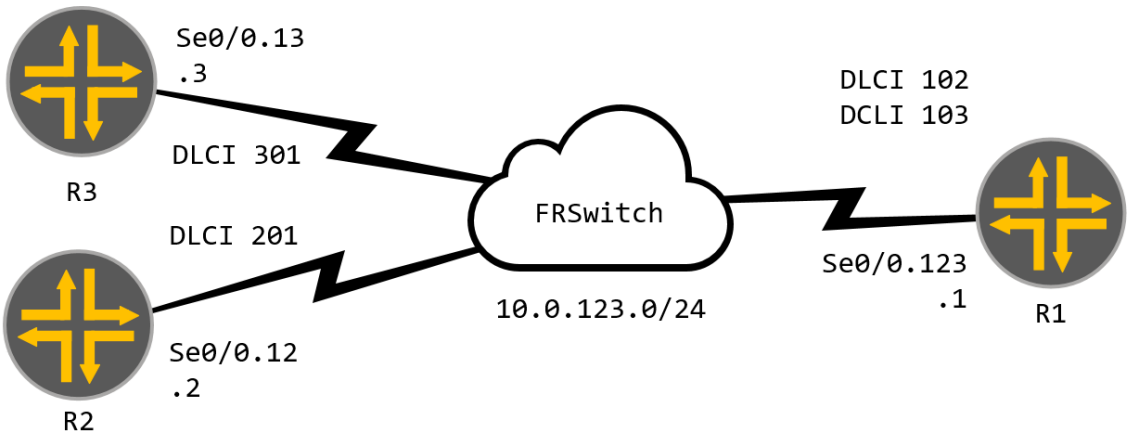
!----------!!----------!!----------!!----------!

```
R1#show frame-relay map
Serial4/0 (up): ip 10.0.0.3 dlci 103(0x67,0x1870), static,
              CISCO, status defined, active
Serial4/0 (up): ip 10.0.0.1 dlci 103(0x67,0x1870), static,
              broadcast,
              CISCO, status defined, active
Serial4/0.1 (up): point-to-point dlci, dlci 113(0x71,0x1C10), broadcast
          status defined, active
Serial4/0.2 (up): point-to-point dlci, dlci 123(0x7B,0x1CB0), broadcast
          status defined, active
Serial4/0.3 (up): point-to-point dlci, dlci 133(0x85,0x2050), broadcast
          status defined, active
```

# Multipoint

## Frame-Relay with Multipoint Sub-interfaces

- If sub-interfaces are being used, the DLCI needs to be linked to the sub-interface
- Inverse ARP configuration is not inherited by sub-interfaces
- With PTP mode sub-interfaces are configured on the hub, each to single location
- With PTMP mode a single interface is configured on the hub to multiple locations
    - Disable **ip split-horizon eigrp <AS>** if using EIGRP, because the update has to travel out of the same interface it came in
- The **no inverse-arp** configuration on the physical interface is not inherited by sub-interfaces




**R1**

```
int se0/0
 encapsulation frame-relay
 no frame-relay inverse-arp
 no shutdown

int se0/0.123 multipoint
 no frame-relay inverse-arp
 ip address 10.0.123.1 255.255.255.0
 frame-relay map ip 10.0.123.3 103 broadcast
 frame-relay map ip 10.0.123.2 102 broadcast

 no ip split-horizon eigrp 123

router eigrp 123
 no auto-summary
 network 0.0.0.0
```

**FRSwitch**

```
frame-relay switching
int s0/1
 encapsulation frame-relay
 frame-relay intf-type dce

int s0/2
 encapsulation frame-relay
 frame-relay intf-type dce

int s0/3
 encapsulation frame-relay
 frame-relay intf-type dce

connect R1-R2 se0/1 102 se0/2 201
```

```
connect R1-R3 se0/1 103 se0/3 301
```

**R2 (R3)**

```
int se0/0
 encapsulation frame-relay
 no frame-relay inverse-arp
 no shutdown

int se0/0.12 point-to-point
 no frame-relay inverse-arp
 ip address 10.0.123.2 255.255.255.0
 frame-relay interface-dlci 102

router eigrp 123
 no auto-summary
 network 0.0.0.0
```

# Authentication

## Frame-Relay Authentication

- Use point-to-point sub-interfaces configured with PPP encapsulation.
- Use ip unnumbered in order to enable self-ping.

**R1 configuration**

```
username R3 password cisco
int se1/0.103 point-to-point
 frame-relay interface-dlci 103 ppp virtual-template 1

interface virtual-template 1
 ip address 10.0.13.1 255.255.255.0
 encapsulation ppp
 ppp authentication chap
 ppp pap sent-username R1 password cisco
```

**R3 configuration**

```
username R1 password cisco
int s1/1.301 point-to-point
 frame-relay interface-dlci 301 ppp virtual-template 1

interface virtual-template 1
 ip address 10.0.13.3 255.255.255.0
 encapsulation ppp
```

```
ppp chap hostname R2
ppp chap password cisco
ppp authentication pap
```

# IOS Services

# Archival

Archive Log
Configure archiving and optionally log commands to syslog

```
archive
 log config
  logging enable
  ;notify syslog
  exit
alias exec sal show archive log config all provisioning
```

Archive Config
Configure archiving of configs to TFTP server

```
archive
 path tftp://192.168.10.1/archive

archive config

show archive
show archive config differences
```

# BFD

## Bidirectional Forwarding Detection (BFD)

- Requires CEF, sent unicast to UDP 3784.
- Only supports asynchronous mode, must be enabled on both sides.
- Only works for directly connected neighbors, BFD itself has no neighbor detection.
- Is not tied to any routing protocol, and can be used as a generic and consistent failure detection mechanism.
- Parts of BFD can be distributed to the data plane (echo), better than reduced IGP timers that exist only at the control plane.

## BFD Echo Mode

- BFD echo packets are are sourced from UDP 3785 and sent to 3785.

- Enabled by default and can be enabled on either side. Does not work alongside ip redirects or uRPF (or IPv6 on CSR).
- Echo mode is supported on single-hop only. The packets are sent on the negotiated BFD timer interval.
- BFD packets are processed in fast switching instead of the control plane.
- Control plane packets are still sent but they are transmitted at the slow timers speed (1000 ms by default).

## BFD Timers

- The time at which 'hello' messages are sent is configured with the interval timer.
- The min_rx timer is the receive timer, if no message is received within this time the neighbor is considered timed-out
- The multiplier specifies how many BFD messages can be missed before neighbor interface is considered down.
- BFD timers work like EIGRP. The send and receive timer do not have to match on both sides
- The slower receive timer of the neighbor will decide the value of the local send timer.

```
bfd slow-timers 1000
int gi0/0
 bfd echo
 bfd interval 500 min_rx 500 multiplier 3
```

## BFD Authentication

```
key chain BFD_KEY
 key 1
  key-string cisco

bfd-template single-hop BFD
 echo
 interval both 500 multiplier 3
 authentication md5 keychain BFD_KEY

int gi0/0
 bfd template BFD
```

## BFD Static

- Static routes that support BFD must specify an egress interface in single-hop mode.
- The neighbor must point back with a static route, or an unassociated route.
- Static routes can be dependent on a group. If one location becomes inaccessible the depending (passive) routes are also removed from the routing table.

**R1**

```
ip route static bfd gi0/0 10.0.12.2
ip route 0.0.0.0 0.0.0.0 gi0/0 10.0.12.2
```

**R2**

```
ip route static bfd gi0/0 10.0.12.1 unassociate
```

**BFD Static Groups**

```
ip route static bfd gi0/0 10.0.12.2 group BFD

ip route 33.33.33.0 255.255.255.0 gi0/1 10.0.13.3
ip route static bfd gi0/1 10.0.13.3 group BFD passive
```

# CEF

## Packet Switching (Routing) Methods

| **Method** | **Description** | **Enable with** |
|---|---|---|
| Process switching | Uses the CPU for each decision on how to forward a packet (CPU intensive, suboptimal)<br>Used in legacy routers and when ACLs are used to match/filter traffic | - |
| Fast Switching | The first packet in a flow is process switched, afterwards the forwarding information is cached<br>Subsequent packets in the flow are forwarded based on the cache | Default<br>**ip route-cache** on interface |
| Cisco Express Forwarding (CEF) | Does not require a CPU lookup of the first packet in a flow<br>Forwarding information is based on FIB and Adjacency table | Default<br>**ip route-cache cef** on interface<br>**ip cef** globally |

Other types of CEF:

- Accelerated CEF (aCEF) - Operation of CEF are distributed (partially) across multiple L3 forwarding engines
    - Other engines cannot load the full FIB, only a portion which is cached
- Distributed CEF (dCEF) - Operation of CEF are distributed (fully) across multiple L3 forwarding engines

**Fast Switching**

```
int gi0/0
 ip route-cache

show ip cache
```

**CEF Switching**

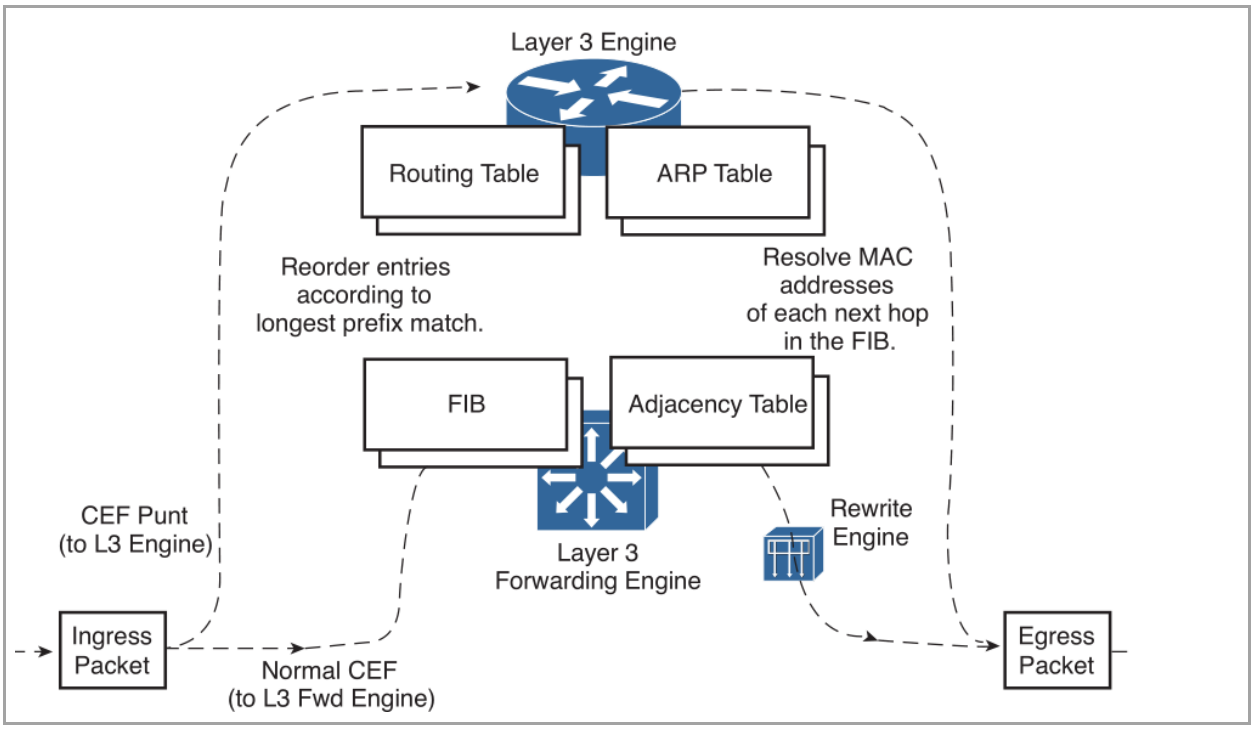
```
ip cef
int gi0/0
 ip route-cache cef
```

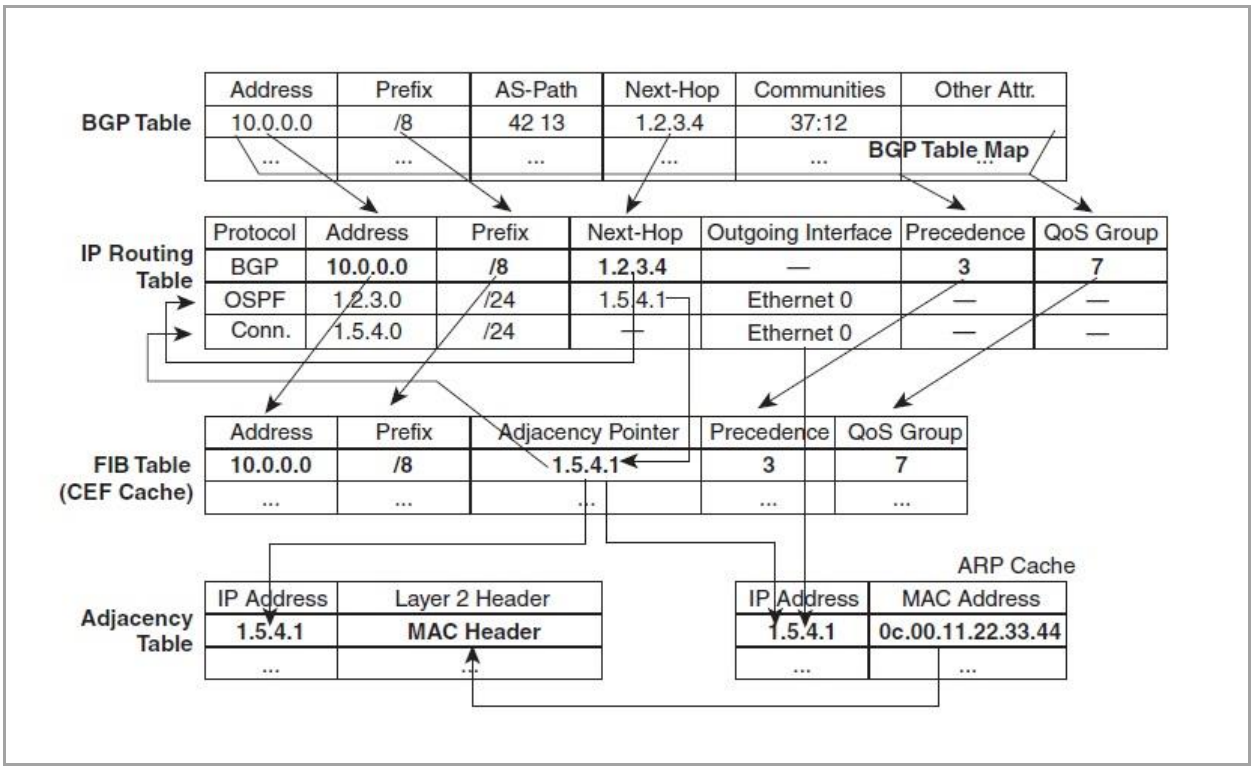
**Verify**

```
R1#show ip interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 10.0.12.1/30
  ...
  IP fast switching is enabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP CEF switching turbo vector
  IP CEF turbo switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
```

## Cisco Express Forwarding

- The routing table is also called the routing information base or RIB
- CEF consists of the Forwarding Information Base (FIB) and the Adjacency Table
    - FIB - Contains the next-hop and reachability information. Stores destination prefixes and egress interfaces
    - Adjacency Table. Protocols like ARP build the adjacency table, which is neighbor IP address information
- After the adjacency and the FIB are built. The RIB is no longer used to route packets, only when a destination is unknown
- The **clear ip route** command clears the FIB and the RIB




- The routing protocol RIB builds the Routing table RIB
- The RIB builds adds a pointer to a route and builds the FIB
- The FIB uses the pointer for the route and adds a next-hop destination address and egress interface

- Protocols like arp and the FIB build the adjacency table that links the pointer to the L2 mac-header and neighbor address

## Adjacency Table

- Consists of both ip- and mac-addresses alongside the outgoing ports/VLANs
  - The mac-address is not shown in the usual format, instead is included in hexadecimal string (see pictures below)
  - Remember that this adjacency table is the other side of the connection (information learned through ARP)
- Adjacencies are added through manual (P2P interface) or dynamic discovery (ARP)
- **Incomplete** adjacencies are caused by:
  - Router cannot use ARP. **clear ip arp** or **clear adjacency** fails to clear the entry
  - MPLS is used but CEF has been administratively disabled (**no ip cef**) or on the interface with **no ip route-cache cef**
- If CEF cannot find a valid adjacency, the packet is sent to the CPU for ARP resolution

## CEF Adjacency Types

| **Adjacency** | **Action** | **Description** |
|---|---|---|
| Null adjacency | Dropped | Packets destined for Null0 |
| Drop adjacency | Dropped | Dropped due to missing route or offline interface<br>Other reasons might be encapsulation failure or unsupported protocols<br>Packets are dropped, but the prefix is checked |
| Discard adjacency | Discarded | Purposefully discarded due to an ACL for example |
| Glean adjacency | Forwarded | Router is directly connected to several hosts<br>FIB holds a subnet prefix instead of host prefix<br>The subnet prefix points to a glean adjacency |
| Punt adjacency | Forwarded | Requires special handling or features that are not yet supported in CEF<br>Forwards the packet to the CPU for process switching |
| Cached Adjacency | Acknowledgement | Update received for the adjacency packet sent |

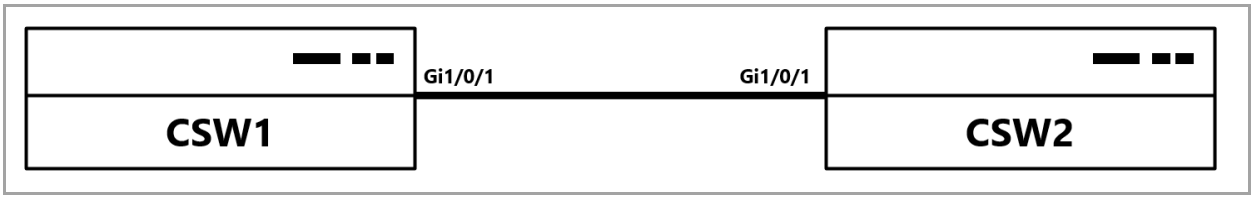
Possible reasons for punting packets:

- An entry cannot be located in the FIB
- The FIB table is full
- The IP Time-To-Live (TTL) has expired
- The maximum transmission unit (MTU) is exceeded, and the packet must be fragmented

- An Internet Control Message Protocol (ICMP) redirect is involved
- The encapsulation type is not supported
- Packets are tunneled, requiring a compression or encryption operation
- An access list with the log option is triggered
- A Network Address Translation (NAT) operation must be performed

CEF Glean

- Used when an ARP entry does not exist
- The packet is forwarded to the routing engine, which generates an arp request (gleans for the mac-address of the remote host)
- During the ARP lookup, the traffic from the source-host is dropped, this is known as ARP throttling or throttling adjacency
    - This throttling lasts for 2 seconds, at which point another ARP request is sent, or if the host replies traffic is forwarded normally




**CSW1**

```
interface gi1/0/1
 switchport mode trunk
 switchport trunk encapsulation dot1q
 mac-address 0000.0000.0001
 no shutdown

interface vlan 10
 ip address 10.0.12.1 255.255.255.0
 mac-address 1010.1010.1010
 no shutdown
```

**CSW2**

```
interface gi1/0/1
 switchport mode trunk
 switchport trunk encapsulation dot1q
 mac-address 0000.0000.0002
 no shutdown

interface vlan 10
 ip address 10.0.12.2 255.255.255.0
 mac-address 2020.2020.2020
```

```
no shutdown
```

**Adjacency output (ping source 10.0.12.1 > 10.0.12.2)**

```
show adjacency
Protocol Interface                 Address
IP       Vlan10                    10.0.12.2(7)
```

```
show adjacency detail
Protocol Interface                 Address
IP       Vlan10                    10.0.12.2(7)
                                   0 packets, 0 bytes
                                   epoch 0
                                   sourced in sev-epoch 2
                                   Encap length 14
                                   202020202020101010101010100800
                                   L2 destination address byte offset
                                   L2 destination address byte length
                                   Link-type after encap: ip
                                   ARP
```

The above hexadecimal value 202020202020101010101010100800 is:

- The first 6 bytes (202020202020) is the mac-address of CSW2 VLAN10 interface (the destination of the ping)
- The second 6 bytes (101010101010) is the mac-address of CSW1 VLAN10 interface (the source of the ping)
- The last 2 bytes (0800) is the ether-type value, which is always 0800 for IPv4

The outgoing interface in this case is shown as VLAN10, there is no reference to gi1/0/1 anywhere in the adjacency table

- The **show ip cef** (see below) also only shows the VLAN

The epoch number denotes the number of times the CEF table has been flushed and regenerated as a whole

- Also has to do with usage of CEF over different line-cards and routing engines (aCEF / dCEF)

```
show adjacency summary
Adjacency table has 1 adjacency:
  each adjacency consumes 308 bytes (0 bytes platform extension)
  1 complete adjacency
  0 incomplete adjacencies
  1 adjacency of linktype IP
    1 complete adjacency of linktype IP
    0 incomplete adjacencies of linktype IP
    0 adjacencies with fixups of linktype IP
    1 adjacency with IP redirect of linktype IP
    0 adjacencies post encap punt capable of linktype IP

Adjacency database high availability:
  Database epoch:        0 (1 entry at this epoch)

Adjacency manager summary event processing:
 Summary events epoch is 2
 Summary events queue contains 0 events (high water mark 1 event)
```

**CEF output**

```
show ip cef 10.0.12.2/32 detail
10.0.12.2/32, epoch 0, flags attached
  Adj source: IP adj out of Vlan10, addr 10.0.12.2 ED255CD8
   Dependent covered prefix type adjfib cover 10.0.12.0/24
  attached to Vlan10
```

# CPPr

## IOS Control Plane

- Handles packets that are not CEF switched, meaning the CPU takes time to handle these packets.
- Maintains keep-alives for routing adjacencies.
- Handles traffic directed at the device itself (management traffic).

## Control Plane Protection (CPPr)

- Framework that consists of traffic classifiers, protection and policing.
- Improvement over Control Plane Policing (CoPP) by allowing finer policing granularity.
- Management Plane Protection (MPP) is a part of CPPr and is basically just specifying a management-interface.
- Depends on CEF. When disabled, CPPr is disabled on sub-interfaces but not on the aggregate interface.

## Control Plane Interfaces

- Host. Handles traffic destined for the router or one of its own interfaces (MGMT, EIGRP, iBGP)
- Transit. Handles software switched IP traffic.
- CEF-Exception. Handles non-IP related packets such as OSPF, eBGP, ARP, LDP and CDP (or packets with TTL <=1) .
- Aggregate interface <cr>. Configuration applied here applies to all the sub-interfaces.

- It is not possible to apply a L3 policy-map to the aggregate and any of the other interfaces at the same time.
- A L3 policy-map applied to the control plane can only use police or drop, not shape...etc.
- The port-filter keyword polices packets going to closed/non-listening TCP/UDP ports.
- The queue-threshold keyword limits the number of protocol packets that are allowed in the input queue.
    - Rate limit OSPF and eBGP on the cef-exception sub-interface, iBGP on the host and EIGRP on the aggregate.

**Police all ICMP traffic**

```
ip access-list extended ICMP_ACL
 permit icmp any any

class-map match-all ICMP_CM
 match access-group name ICMP_ACL
policy-map ICMP_PM
 class ICMP_CM
  police 10000 conform-action transmit exceed-action drop

control-plane host
 service-policy input ICMP_PM
```

**Drop connections to closed ports**

```
class-map type port-filter match-all CLOSED_PORTS_CM
 match closed-ports

policy-map type port-filter CLOSED_PORTS_PM
 class CLOSED_PORTS_CM
  drop

control-plane host
 service-policy type port-filter input CLOSED_PORTS_PM
```

**Queue SNMP traffic to 75 and any other open UDP/TCP ports to 100**

```
class-map type queue-threshold SNMP_CM
 match protocol snmp
class-map type queue-threshold HOST_CM
 match host-protocols

policy-map type queue-threshold QUEUE_PM
```

```
class SNMP_CM
 queue-limit 75
class HOST_CM
 queue-limit 100

control-plane host
 service-policy type queue-threshold input QUEUE_PM
```

**Rate limit EIGRP traffic (requires egress direction)**

```
ip access-list extended EIGRP
permit eigrp any any

class-map match-all EIGRP_CM
 match access-group name EIGRP

policy-map EIGRP_PM
 class EIGRP_CM
  police 10000 conform-action transmit exceed-action drop

control-plane
 service-policy output EIGRP_PM
```

## Management Plane Protection (MPP)

Multiple interfaces can be specified for different protocols

```
control-plane host
 management-interface fa0/0 allow ssh
 management-interface fa0/1 allow snmp
```

The following protocols can be allowed:

- beep - Beep Protocol
- ftp - File Transfer Protocol
- http - HTTP Protocol
- https - HTTPS Protocol
- snmp - Simple Network Management Protocol
- ssh - Secure Shell Protocol
- telnet - Telnet Protocol
- tftp - Trivial File Transfer Protocol
- tl1 - Transaction Language Session Protocol

# EEM

## Embedded Event Manager (EEM)

- The skip keyword prevents the command from being executed. Default is skip no.

- The sync keyword runs the script before the command. Default is sync yes.
- _exit_status 1 means that the command is run.
- _exit_status 0 means that the command is skipped.
- $_cli_result is the outcome of a cli command that was executed, can be pasted into the console with the puts keyword.
- $_cli_msg is the pattern matched with the event keyword. Can be pasted into the console with the syslog msg keyword.
- The cli command is not executed until the EEM policy exits.

```
show event manager policy registered
debug event manager action cli
debug event manager action mail
```

## EEM Examples

**Disable show running-config command:**

```
event manager applet DIS_SH_RUN
 event cli pattern "show run" skip yes sync no
 action 1.0 cli command "enable"
 action 1.1 syslog msg "$_cli_msg not executed, function disabled"
 action 1.2 mail server ....
```

**Hide interfaces from the running configuration:**

```
event manager applet SH_RUN_NO_INT
 event cli pattern "show run" sync yes
 action 1.0 syslog msg "$_cli_msg executed"
 action 1.1 cli command "enable"
 action 1.2 cli command "show run | section exclude interface"
 action 1.3 puts "$_cli_result"
```

**Re-enable manually shut down interfaces:**

```
event manager applet NO_SHUT_INT
 event syslog pattern "Interface FastEthernet0/0, changed state to administratively down"
 action 1.0 cli command "enable"
 action 1.1 cli command "configure terminal"
 action 1.2 cli command "interface Fa0/0"
 action 1.3 cli command "no shut"
```

**Print confirmation to the terminal:**

```
event manager applet WRITE_MEMORY
 event cli pattern "write memory" sync yes
 action 1.0 syslog msg "$_cli_msg Command Executed"
 set 2.0 _exit_status 1
```

**Disable OSPF and EIGRP:**

```
event manager applet DIS_OSPF_EIGRP
 event cli pattern "router [eEoO].*" sync no skip yes
 action 1.0 syslog msg "Routing protocols OSPF and EIGRP have been disabled"
```

**Send ICMP requests based on tracking object:**

```
event manager applet TRACK_1_DOWN
 event syslog pattern "1 ip sla 1 state Up->Down"
 action 1.0 syslog msg "IP SLA 1 Transferred to Down State, Testing ICMP"
 action 1.1 cli command "enable"
 action 1.2 cli command "ping 10.0.12.2 repeat 5 time 1"
 action 1.3 syslog msg "ping 10.0.12.2 repeat 5 time 1"
 action 1.4 puts "$_cli_result"

event manager applet TRACK_1_UP
 event syslog pattern "1 ip sla 1 state Down->Up"
 action 1.0 syslog msg "IP SLA 1 Returned to UP State, Testing ICMP"
 action 1.1 cli command "enable"
 action 1.2 cli command "ping 10.0.12.2 repeat 5 time 1"
 action 1.3 syslog msg "ping 10.0.12.2 repeat 5 time 1"
 action 1.4 puts "$_cli_result"
```

**Or:**

```
event manager applet TRACK_1_DOWN
 event track 1 state down
 ...
```

**Periodically send output to the console:**

```
ip sla 1
 udp-jitter 192.168.0.2 16384 codec g729a
  frequency 5

event manager applet IP_SLA_1
 event timer watchdog time 3600
 action 1.0 cli command "show ip sla statistics 1"
 action 1.2 puts "$_cli_result"
 action 1.3 mail ...
```

**Create a log message based on added routes (does not disable function):**

```
event manager applet STATIC_ROUTES
 event routing type add protocol static network 0.0.0.0/0  le 32
 action 1.0 syslog msg "Static routes are not allowed, notifying admin"
 action 1.1 mail server ....
```

# EPC

**Old Method**
Association (and disassociation) are actions that can be performed in order to bind a capture point to a capture buffer.

- A capture point can only be associated with one capture buffer (an ACL filter can also be applied).
- A capture buffer can be associated with many capture points.
- A buffer can collect data from many points but a point can send data to only one buffer.
- Capture local traffic with the monitor capture point ip process-switched LOCAL from-us command.

```
monitor capture buffer BUFFER
monitor capture point ip cef PCAP fa0/0 both
monitor capture point associate PCAP BUFFER

show monitor capture buffer BUFFER dump
show monitor capture buffer BUFFER parameters
show monitor capture point PCAP

monitor capture buffer BUFFER export
```

**New Method**

```
monitor capture PCAP match any int gi0/0 both
monitor capture PCAP start

show monitor capture PCAP

monitor capture PCAP export
```

# KRON

Wednesday, October 12, 2016
2:31 PM

## Command Scheduler (KRON)

- Only works for exec mode commands, not global or interface configuration commands
- Choose either the oneshot or recurring keyword to schedule KRON occurrence once or repeatedly
- The system-startup keyword will set the occurrence to be at system startup

**Show routes every 5 minutes:**

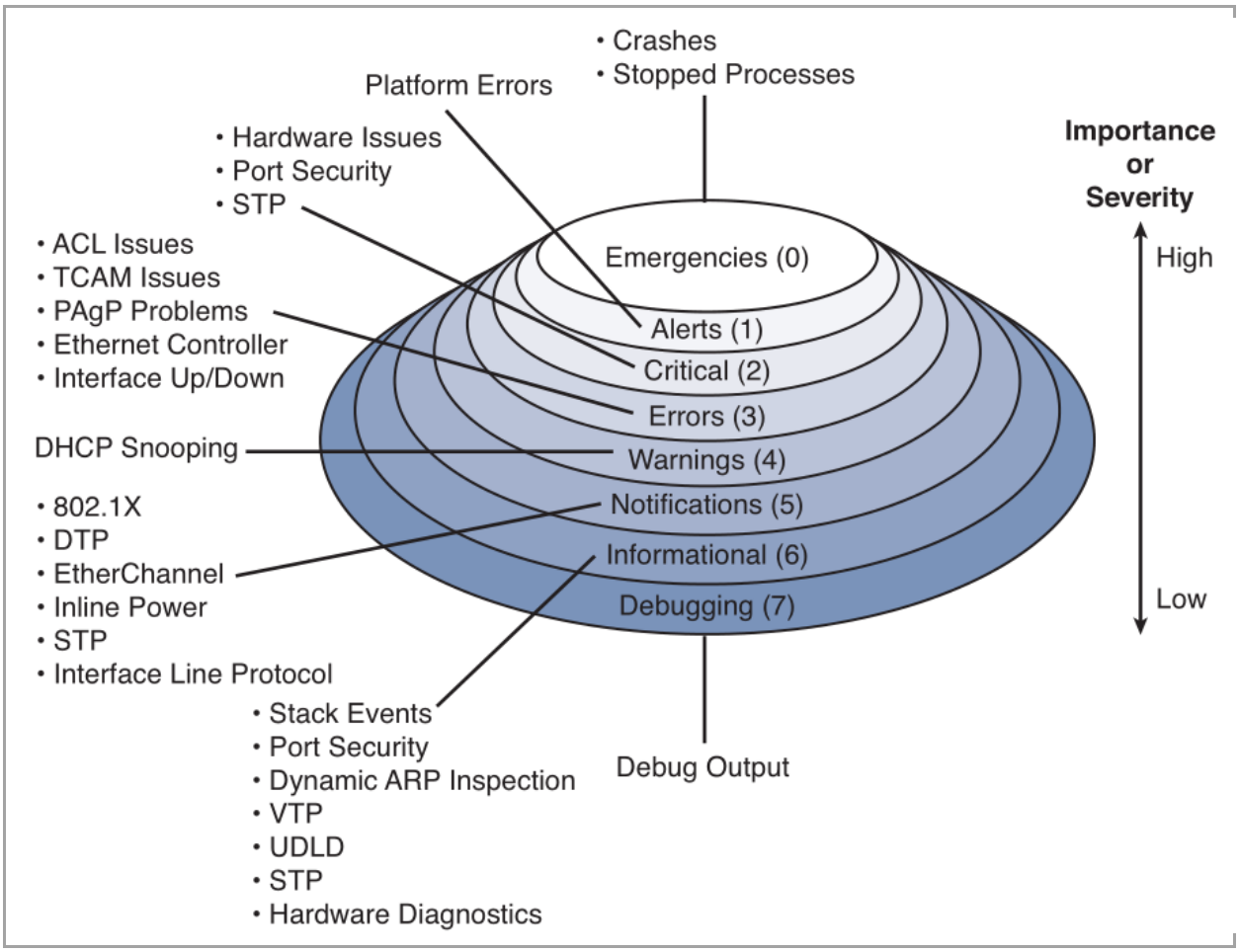
```
kron policy-list KRON_POLICY
 cli show ip route

kron occurrence KRON_OCC in 5 recurring
 policy-list KRON_POLICY
```

```
sh kron schedule
debug kron all
```

# Logging & Syslog

## Logging Severity Levels

| Level Keyword | Level | Description | Syslog Definition |
|---|---|---|---|
| emergencies | 0 | System unstable | LOG_EMERG |
| alerts | 1 | Immediate action needed | LOG_ALERT |
| critical | 2 | Critical conditions | LOG_CRIT |
| errors | 3 | Error conditions | LOG_ERR |
| warnings | 4 | Warning conditions | LOG_WARNING |
| notifications | 5 | Normal but significant condition | LOG_NOTICE |
| informational | 6 | Informational messages only | LOG_INFO |
| debugging | 7 | Debugging messages | LOG_DEBUG |




## Logging Destinations

| Destination | Location | Enabled by default |
|---|---|---|
| Buffer | Local buffer | yes |
| Console | Local console | yes |
| Monitor | VTY lines | no |
| Host | Remote syslog | no |

Disable all logging capabilities (except the console) with the **no logging on** global command

- Output of **show logging** will display %SYS-5-LOGGING_STOP: Logging is disable - CLI initiated

**Buffer**

- Default severity is 7 (debugging)
- Enable the buffer with the logging buffered global command (enabled by default / might be disabled in the lab)

    - The default size differs per platform, according to the CCNP guide it is 4096 bytes by default (roughly 50 lines of text)
- When the buffer is full, the oldest messages will be overridden (circular)

```
logging buffered 6
logging buffered informational
logging buffered 4096
```

**informational** - You can configure the severity level by the number (6) or by name (informational)

**Console**

- Default severity is 7 (debugging)
- By default, syslog information sent to the console is only shown when connected to the console port
- This is not very efficient way to collect and view system messages because of its low throughput
    - Logging to a VTY line is more effective (requires logging monitor)

```
logging console 7
logging console guaranteed
```

**guaranteed -** This keyword ensures that all debugging messages will be sent to the console, even if a lot of output is generated

- This guaranteed delivery is achieved by slowing down other system processes and preferring the logging process
- Can lead to other time-critical processes failing

**Monitor**

- More effective than console because of line speed and performance
- Disabled by default

```
logging monitor 5
```

**Host**

- Remote syslog server (disabled by default)
- Default transport protocol is UDP on port 514
- The logging trap defines the severity level of remote syslog logging
    - Default severity is informational (6)

**logging host** *ip-address* **transport** *udp* | *tcp* **port** *port-id*

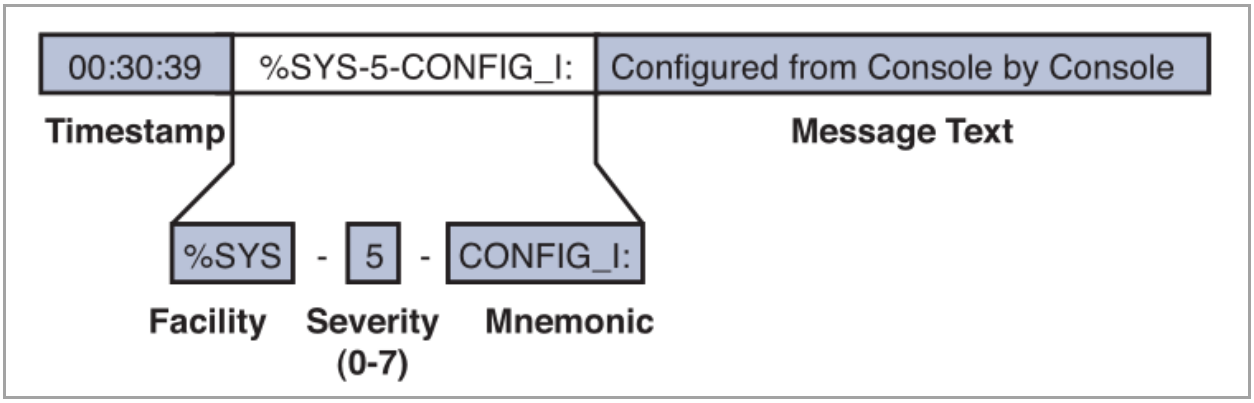
```
logging host 1.1.1.1
logging trap 6
```

```
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 8 flushe
No Active Message Discriminator.
No Inactive Message Discriminator.
    Console logging: level debugging, 329 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level notifications, 120 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level informational, 21 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled
    Trap logging: level informational, 33 message lines logged
        Logging to 1.1.1.1  (udp port 514, audit disabled,
              link up),
              1 message lines logged,
              0 message lines rate-limited,
              0 message lines dropped-by-MD,
              xml disabled, sequence number disabled
              filtering disabled
        Logging Source-Interface:       VRF Name:
```

## Logging Message Format

| **Element** | **Format** | **Purpose** | **Enabled** | **Command** |
|---|---|---|---|---|
| Sequence # | xxxxxx | Simple incrementing number | no | service sequence-numbers |
| Timestamp | HH:MM:SS<br>M D<br>HH:MM:SS | Device uptime using internal clock<br>Actual time using internal clock or NTP | yes<br>no | service timestamps log uptime<br>service timestamps log datetime |
| Facility | % | Categorizes the function or module that generated the message<br>Default is Local7 | yes | |
| Severity | 0-7 | Syslog severity level | yes | |

| | | | | |
|---|---|---|---|---|
| MNEMONIC | CONFIG_I | Text string that uniquely describes the message | yes | |
| Description | ... | Text string containing detailed information about the event | yes | |




**Timestamp**

- The device uptime value is used as the timestamp by default, this is not very helpful
- Can be changed to use the current time/date value (local/NTP)

| |
|---|
| **service timestamps log datetime** *localtime* \| *show-timezone* \| *msec* \| *year* |

**localtime** - Display time on log message using the local time-zone, otherwise UTC is assumed
**show-timezone** - Displays the configured timezone in the log message
**msec** - Display time on log message in milliseconds
**year** - Display the current year in the log message

## Logging Filtering

- Many messages can be generated on switches (sent to either the buffer or a syslog host)
- Apply filtering with the **no logging event** global command
- An example is the link-state protocol of a port (up/down), which can occur often on access switches
    - Disable link-state protocol logging with **no logging event link-status**

## Logging Discriminator

- Filter certain messages from being logged to a destination
- Can apply filtering based on facility, severity level, contents of the message or all at the same time
- Can only apply a single discriminator per destination (console, buffer, host, monitor)

**Filter syslog message to console when exiting configuration mode**

```
logging discriminator CONSOLE msg-body drops Configured from console by console
logging console discriminator CONSOLE
```

```
show logging
Active Message Discriminator:
CON       msg-body       drops    Configured from console by console

    Console logging: level debugging, 13 messages logged, xml disabled,
                     filtering disabled, discriminator(CON),
                     0 messages rate-limited, 1 messages dropped-by-MD
```

**Filter syslog message severity 3 to buffer when link goes up down**

```
%LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1, changed
```

```
logging discriminator LINK severity drops 3 msg-body drops UPDOWN
logging buffered discriminator LINK
```

```
show logging
Active Message Discriminator:
LINK      severity group drops    3
          msg-body       drops    UPDOWN

    Buffer logging:  level debugging, 0 messages logged, xml disabled,
                     filtering disabled, discriminator(LINK),
                      0 messages rate-limited, 2 messages dropped-by-MD
```

## Logging Rate Limit

- Limit the amount of log messages that are generated per second
- Can limit the rate-limiting specifically to the console line
    - Cannot rate-limit the buffer, host or terminal only
    - Rate-limit either all destinations or the console only

**logging rate-limit** *messages-per-second* **console**
**logging rate-limit** *messages-per-second* **all except** *severity-level*

**all** - Rate-limit all messages (including debugs)
**except** - Exclude certain severity levels from the rate-limiting

## Logging History

- The logging history (not the command history) is a buffer that stores syslog messages sent via SNMP
    - Enable sending of syslog messages to a SNMP NMS with the **snmp-server enable traps syslog** global command
- Syslog messages sent via SNMP can be lost in transit, for this reason the recent logging messages are also stored in the logging history table
    - The logging history buffer is enabled even if SNMP is not configured
    - The default severity level is 5
    - The default size is 1 (meaning a single syslog message)
    - New messages will override older messages (circular)

```
logging history size 1-500
logging history 5
```

```
show logging history
Syslog History Table:1 maximum table entries,
saving level warnings or higher
 41 messages ignored, 0 dropped, 0 recursion drops
 3 table entries flushed
 SNMP notifications not enabled
   entry number 4 : LINK-3-UPDOWN
    Interface Ethernet0/2, changed state to up
    timestamp: 149048
```

## Command history

- This is the actual command history
- If you disable the history (no history) you also disable going to the previous commands (up arrow)
- Default history size is 20 commands

```
line con0
 history
 history size 20
```

```
terminal history size 20
show history
```

**Reset command history**

```
term history size 0
term history size 20
```

# NetFlow

## NetFlow Versions

| Version | Description |
|---|---|
| 5 | NetFlow v5 is fixed format, cannot be extended or added to<br>IPv4 only<br>Added BGP AS information and sequence numbers<br>Exports data from main cache only |
| 8 | Added support for data export from aggregation caches |
| 9 | NetFlow v9 can add additional information to flows, template based<br>Added support for MPLS, BGP next-hop and IPv6 headers<br>Exports data from main and aggregation cache |

## NetFlow IP Flows

- NetFlow Requires CEF in order to function
- In original NetFlow if all of characteristics match, they're considered the same flow
- An IP Flow can be characterized by a set of 5 and up to 7 packet attributes:
    - Source / destination IP address
    - Source / destination port
    - L3 protocol type
    - Class of Service
    - Router or switch interface

## NetFlow Collector

- NetFlow Collector = NetFlow server
- Flow records store IP flow information
- The Collection Engine (local) exports NetFlow data to the collector with 1.5% export data overhead
    - The NetFlow Cache creates cache entries (flow records) for every active flow
    - NetFlow export, unlike SNMP polling, pushes information periodically to the collector
    - Flows that have terminated or expired (based on cache) are exported as well

```
ip flow-export destination 10.0.0.100 9995
ip flow-export version 9
ip flow-export source loopback 0
```

- Flows can be exported using version 1, 5 or 9
    - Version 9 exports data from main and aggregation cache

```
show ip flow export
Flow export v9 is enabled for main cache
  Export source and destination details :
```

```
VRF ID : Default
  Source(1)      1.1.1.1 (Loopback0)
  Destination(1)  10.0.0.100 (9995)
Version 9 flow records
0 flows exported in 0 udp datagrams
0 flows failed due to lack of export packet
0 export packets were sent up to process level
0 export packets were dropped due to no fib
0 export packets were dropped due to adjacency issues
0 export packets were dropped due to fragmentation failures
0 export packets were dropped due to encapsulation fixup failures
```

## NetFlow v5

- NetFlow v5 does not have a concept of 'ingress' and 'egress' flows
- The collector engine reverses the information behind the scenes without needing any additional configuration
- The **ip route-cache flow** command is the old way of configuring NetFlow
    - This is called the flow fast-switching cache
    - The old command will also enable NetFlow on all sub-interfaces, the newer command **ip flow ingress** does not
    - The old command is actually converted to **ip flow ingress** on newer devices

```
int gi0/0
 description INSIDE
 ip flow ingress
```

**show ip flow interface**

```
GigabitEthernet0/0
  ip flow ingress
```

**show ip cache flow**

```
IP packet size distribution (51 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .686 .019 .294 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456704 bytes
  2 active, 65534 inactive, 5 added
  152 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 533256 bytes
  0 active, 16384 inactive, 0 added, 0 added to flow
```

```
 0 alloc failures, 0 force free
 1 chunk, 1 chunk added
 last clearing of statistics never
Protocol       Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------       Flows     /Sec    /Flow  /Pkt    /Sec     /Flow     /Flow
TCP-Telnet         1     0.0       11   42      0.0       2.2       1.2
ICMP               1     0.0       10   100     0.0       0.2      15.0
IP-other           1     0.0        1   40      0.0       0.0      15.9

SrcIf       SrcIPaddress    DstIf       DstIPaddress    Pr SrcP DstP  Pkts
Total:                  3     0.0        7    68     0.1       0.8      10.7

SrcIf       SrcIPaddress    DstIf       DstIPaddress    Pr  SrcP  DstP   Pkts
Gi0/0       10.0.12.2       Local       1.1.1.1         01 0000 0800    5
Gi0/0       10.0.12.2       Local       10.0.12.1       06 8DD9 0017    10
Gi0/0       10.0.12.2       Null        224.0.0.10      58 0000 0000    24
```

The protocol (Pr), source port (SrcP) and destination port (DstP) in the output above are in Hexadecimal (HEX)

- The second line 06 8DD9 0017 is actually:
  - Protocol HEX 06 > BINARY 0110 > DECIMAL 6 (TCP)
  - Source port HEX 8DD9 > BINARY 1000 1101 1101 1001 > DECIMAL 36313) Random(
  - Destination Port HEX 0017 > BINARY 0000 0000 0001 0111 > DECIMAL 23 (Telnet)
- So the second line, traffic from 10.0.12.2 to the local interface (10.0.12.1) is actually a Telnet session to port 23

## NetFlow v9

- NetFlow v9 introduces the 'egress' flows concept, possible to capture both ingress and egress on same interface
  - Egress calculates flows after compression
  - Ingress calculates before compression, this is a problem if WAN links are using compression of packets
  - Multicast traffic cannot be effectively matched on ingress (see destination interface null above)
- Configuring NetFlow v9 with the **egress** keyword uses a default build-in template behind the scenes
- Only when templates are specified Flexible NetFlow (see below) is being used

```
int gi0/0
 description INSIDE
 ip flow ingress
int gi0/1
 description OUTSIDE
 ip flow egress
```

- Egress flows are marked with * in **show ip cache flow**

**show ip cache flow**

```
IP packet size distribution (1479 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .960 .002 .037 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 4456704 bytes
  3 active, 65533 inactive, 22 added
  6371 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 533256 bytes
  3 active, 16381 inactive, 11 added, 11 added to flow
  0 alloc failures, 0 force free
  1 chunk, 4 chunks added
  last clearing of statistics never
Protocol       Total    Flows   Packets Bytes  Packets Active(Sec) Idle(Sec)
--------       Flows     /Sec     /Flow  /Pkt     /Sec     /Flow     /Flow
TCP-Telnet         5     0.0        12    42     0.0       1.0       4.2
ICMP               6     0.0         9   100     0.0       0.4      15.5
IP-other           8     0.0       162    59     0.2     746.2       7.3

SrcIf       SrcIPaddress    DstIf       DstIPaddress    Pr SrcP DstP  Pkts
Total:           19     0.0        74    60     0.2     314.6       9.1

SrcIf       SrcIPaddress    DstIf       DstIPaddress    Pr SrcP DstP  Pkts
Gi0/0       10.0.12.2       Gi0/1*      10.0.15.5       06 33B3 0017    11
Gi0/0       10.0.12.2       Gi0/1*      10.0.15.5       01 0000 0800    15
Gi0/0       10.0.12.2       Gi0/1       10.0.15.5       06 33B3 0017    11
Gi0/0       10.0.12.2       Null        224.0.0.10      58 0000 0000    45
```

## NetFlow Aggregation Cache (v8 and v9)

- Enables specification of which type of flows will be exported to the collector
- All flows are still captured on the device (using v5 or v9)
    - Only when flows are exported they are combined (aggregated) using v8 or v9 (v5 is not possible)

**Only export flow entries that have a /32 mask**

```
ip flow-aggregation cache destination-prefix
 cache entries 1024
 export version 9
 export destination 2.2.2.2 9995
 mask destination minimum 32
```

enabled

**show ip cache flow aggregation destination-prefix**
IP Flow Switching Cache, 69636 bytes
  1 active, 1023 inactive, 1 added
  2 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 9096 bytes
  1 active, 255 inactive, 1 added, 1 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added

Minimum destination mask is configured to /32

| Dst If | Dst Prefix | Msk | AS | Flows | Pkts | B/Pk | Active |
|---|---|---|---|---|---|---|---|
| Local | 1.1.1.1 | /32 | 0 | 1 | 538 | 100 | 16.8 |

**show ip flow export**
Flow export v1 is disabled for main cache
  Version 1 flow records
  Cache for destination-prefix aggregation v8
  VRF ID : Default
    Destination(1)  10.0.0.100 (9995)
    Minimum destination mask configured to /32
  2 flows exported in 2 udp datagrams
  0 flows failed due to lack of export packet
  0 export packets were sent up to process level
  2 export packets were dropped due to no fib
  0 export packets were dropped due to adjacency issues
  0 export packets were dropped due to fragmentation failures
  0 export packets were dropped due to encapsulation fixup failures

# FNF

## Flexible NetFlow (FNF)

Consists of three parts:

- Flow Records, which set key and non-key fields
- Flow Exporter, which details where and how to send the exports
- Flow Monitors, which match the flow records and exporters, and are then applied to an interface

```
flow exporter FNF_EXPORT
 destination 10.0.0.100
 transport udp 9995
 export-protocol netflow-v9
 source loopback0

flow monitor FNF
 record netflow ipv4 original-input
 exporter FNF_EXPORT

int gi0/0
 ip flow monitor FNF input
```

**show flow interface**

```
Interface GigabitEthernet0/0
  FNF:  monitor:          FNF
        direction:        Input
        traffic(ip):      on
```

**show flow exporter**

```
Flow Exporter FNF_EXPORT:
  Description:              User defined
  Export protocol:          NetFlow Version 9
  Transport Configuration:
    Destination IP address: 10.0.0.100
    Source IP address:      1.1.1.1
    Source Interface:       Loopback0
    Transport Protocol:     UDP
    Destination Port:       9995
    Source Port:            63313
    DSCP:                   0x0
    TTL:                    255
    Output Features:        Not Used
```

**show flow monitor FNF cache**

```
  Cache type:                           Normal
  Cache size:                           4096
  Current entries:                      1
  High Watermark:                       1

  Flows added:                          1
  Flows aged:                           0
```

```
   - Active timeout      (  1800 secs)          0
   - Inactive timeout    (    15 secs)          0
   - Event aged                                 0
   - Watermark aged                             0
   - Emergency aged                             0

IPV4 SOURCE ADDRESS:       10.0.12.2
IPV4 DESTINATION ADDRESS:  224.0.0.10
TRNS SOURCE PORT:          0
TRNS DESTINATION PORT:     0
INTERFACE INPUT:           Gi0/0
FLOW SAMPLER ID:           0
IP TOS:                    0xC0
IP PROTOCOL:               88
ip source as:              0
ip destination as:         0
ipv4 next hop address:     0.0.0.0
ipv4 source mask:          /30
ipv4 destination mask:     /0
tcp flags:                 0x00
interface output:          Null
counter bytes:             1980
counter packets:           33
timestamp first:           20:56:01.431
timestamp last:            20:58:32.127
```

## FNF Sampler

```
sampler FNF_SAMPLER
 mode random 1 out-of 10

int gi0/0
 ip flow monitor FNF sampler FNF_SAMPLER input
```

```
show sampler FNF_SAMPLER
Sampler FNF_SAMPLER:
  ID:             3930404548
  export ID:      1
  Description:    User defined
  Type:           random
  Rate:           1 out of 10
  Samples:        1
```

```
Requests:       3
Users (1):
  flow monitor FNF (ip,Gi0/0,Input)  1 out of 3
```

# Sampler / Top Talkers

## NetFlow Top Talkers

- Useful if no collector server is present to analyze data flows
- Shows top talkers (busiest hosts) based on bytes or packets

```
ip flow-top-talkers
 top 10
 sort by packets
```

**show ip flow top-talkers**

| SrcIf | SrcIPaddress | DstIf | DstIPaddress | Pr SrcP DstP | Pkts |
|---|---|---|---|---|---|
| Gi0/0 | 10.0.12.2 | Null | 224.0.0.10 | 58 0000 0000 | 107 |
| Gi0/0 | 10.0.12.2 | Local | 1.1.1.1 | 01 0000 0800 | 57 |

- The protocol (Pr), source port (SrcP) and destination port (DstP) in the output above are in Hexadecimal (HEX)

## NetFlow Sampler

- Sampled mode lets you collect only for a subset of traffic
- Can be linked directly to the interface, or be part of a policy-map
    - Default direction is ingress, use the **egress** keyword for egress
- Cannot be used alongside the ingress command
    - Either capture all flows or a subset of flows

```
flow-sampler-map RANDOM
 mode random one-out-of 10

int gi0/0
 flow-sampler RANDOM egress
```

**show flow-sampler**

```
Sampler : RANDOM, id : 1, packets matched : 1, mode : random sampling mode
 sampling interval is : 10
```

## NetFlow Policy-Map Sampler

```
flow-sampler-map RANDOM
```

```
 mode random one-out-of 10
flow-sampler-map ONE_ONE
 mode random one-out-of 1

class-map match-all ICMP
 match protocol icmp
policy-map SAMPLER
 class ICMP
  netflow-sampler RANDOM
 class class-default
  netflow-sampler ONE_ONE

int gi0/0
 service-policy input SAMPLER
```

```
show policy-map interface
 GigabitEthernet0/0

  Service-policy input: SAMPLER

    Class-map: ICMP (match-all)
      100 packets, 11400 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: protocol icmp
      netflow-sampler: RANDOM

    Class-map: class-default (match-any)
      16 packets, 1184 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
      netflow-sampler: ONE_ONE
```

# NTP

## Network Time Protocol (NTP)

- Masters on older IOS versions (≤12.4) use the 127.127.7.1 local address to peer with itself
- Newer IOS versions use the 127.127.1.1 local address
- Stratum is just a hop-count, less hops is better
  - A hop count of 16 means infinite
  - Default stratum is 8 when just configuring **ntp master <cr>**
- The source local address is always one stratum lower than the configured value
  - Stratum is the tie-breaker. If two servers offer the same stratum, the **prefer** keyword can be added to prefer one over the other

```
ntp master 10
ntp server 10.0.0.1 prefer
ntp peer 10.0.0.2
```

- NTP Peers can act as a client or a server at the same time and offer bidirectional synchronization
    - When the connection to the NTP server fails, the peer will be regarded as the new server

```
show ntp associations detail

10.0.0.1 configured, ipv4, insane, invalid, stratum 5
delay 1.00 msec, offset -0.5000 msec, dispersion 189.45, jitter 0.97 msec
```

```
show ntp status

Clock is unsynchronized, stratum 16
clock offset is 7926.0852 msec, root delay is 36.10 msec
```

The offset value is the time difference in milliseconds between the local clock and the NTP server's reference clock.

- A insane peer is unsynchronized, a sane peer is synchronized
- The offset must be < 1000 msec (1 second) in order for the NTP source/server to be considered sane
- NTP does not shift the clock instantaneously, instead the router slowly drifts towards the time
- If the offset value between the client and the server is large, this process can take a long time
- After the offset value is < 1 second off, the router will adjust its stratum from 16 (infinite) to the appropriate stratum

```
show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   dis
*~10.0.0.1        127.127.1.1      5     52     64   377  1.000   0.500  3.90
 ~10.0.0.2        127.127.1.1      8     67     64   376  0.000   0.000  3.74
+~127.127.1.1     .LOCL.           9      8     16   377  0.000   0.000  1.20
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
show ntp status

Clock is synchronized, stratum 5, reference is 10.0.0.1
```

## Time Zones

- NTP updates are always sent in UTC/GMT
- EU and US summer time dates are different.
    - Default is US, configure with **clock summer-time US recurring**
- US summer time begins second Sunday in March, ends first Sunday in November
- EU summer time begins last Sunday in March, ends last Sunday in October

```
clock timezone CANADA_PACIFIC -8
clock summer-time US recurring 2 Sun Mar 02:00 First Sun Nov 02:00
clock summer-time EU recurring Last Sun Mar 02:00 Last Sun Oct 02:00
```

## NTP Broadcast and Multicast

- Default multicast address is 224.0.1.1
- Defining an interface as broadcast/multicast will stop reception of NTP unicast requests on that interface

Server

```
int gi1/0/1
 description TO_CLIENTS
 ntp broadcast
 ntp multicast 224.0.1.1
```

Client

```
int gi0/0
 ntp broadcast client
 ntp multicast client
```

# Auth / Access

## NTP Authentication

- The client authenticates the server, it is more important to receive time from the correct source over giving time to devices
    - Giving time to specific devices is done through NTP Access Control (see below)
- Other NTP clients will still be able to request time without authentication
- You only need to configure the **ntp authenticate** command on the client

**Server**

```
ntp trusted-key 1
ntp authentication-key 1 md5 cisco
```

**Client**

```
ntp authentication-key 1 md5 cisco
ntp trusted-key 1
ntp authenticate
ntp server 1.1.1.1 key 1
```

## NTP Access Control

- Default behavior is allow NTP access to everyone
- Defining and applying an ACL will implicitly deny all other NTP traffic
- NTP Control messages (QUERIES) are for reading and writing internal NTP variables and status information
    - QUERIES are not used for synchronization, REQUEST and UPDATE messages are used for time synchronization

- Access-groups are applied from most permissive to most restrictive
    - Peer is most permissive, serve-only is most restrictive
    - This means that defining the same host as peer and serve-only, the client will still be able to peer

```
access-list 1 permit host 2.2.2.2

access-list 2 permit host 2.2.2.2

ntp master
ntp peer 2.2.2.2
ntp access-group peer 1
ntp access-group serve-only 2
```

- The **serve-only** keyword allows only time requests from NTP clients
- The **peer** keyword allows bidirectional synchronization, time requests and NTP control queries from clients/peers

**IOS 12.4**

```
access-list 1 permit host 127.127.7.1
access-list 1 permit host 2.2.2.2

ntp master
ntp peer 2.2.2.2
ntp access-group peer 1
```

- Masters on older IOS versions (12.4) need to specifically allow peering with the own loopback address (127.127.7.1)

**show access-lists**
Standard IP access list 1
    10 permit 2.2.2.2 (1 match)
    20 permit 127.127.7.1 (1 match)

**show ntp associations**

| address | ref clock | st | when | poll | reach | delay | offset | disp |
|---|---|---|---|---|---|---|---|---|
| +~2.2.2.2 | 127.127.1.1 | 8 | 47 | 64 | 77 | -1895. | 21623. | 11179. |
| *~127.127.7.1 | 127.127.7.1 | 7 | 8 | 64 | 337 | 0.0 | 0.00 | 125.0 |

 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

# SNTP

Switch(config)# ntp authentication-key key-number md5 key-string
Switch(config)# ntp authenticate
Switch(config)# ntp trusted-key key-number
Switch(config)# ntp server ip-address key key-number

Switch(config)# access-list acl-num permit ip-address mask
Switch(config)# ntp access-group {serve-only | serve | peer | query-only } acl-num

As its name implies, the Simplified Network Time Protocol (SNTP) offers a reduced set of NTP functions. When a switch is configured for SNTP, it operates as an NTP client only. In other words, the switch can synchronize its clock with an NTP server, but it cannot allow other devices to synchronize from its own clock. Time synchronization is also simplified, resulting in a slightly less accurate result.
To configure SNTP, use the typical NTP commands and substitute sntp for the ntp keyword. For example, the following SNTP configuration commands correspond to their NTP counterparts:
Switch(config)# sntp authentication-key key-number md5 key-string
Switch(config)# sntp authenticate
Switch(config)# sntp trusted-key key-number
Switch(config)# sntp server ip-address key key-number

sntp master 10
sntp server 10.0.0.1 prefer
sntp peer 10.0.0.2

# PPP

## PPP Authentication

- The router that enables PPP authentication requests credentials from the remote router.
- The credentials supplied by the remote router has to match the local user database.
- Local authentication is based on usernames.
- EAP requires the addition of the local keyword to authenticate using the local database.
- PPP usernames can still be used for line management. Configure the **username PPP-USER autocommand logout** command to prevent this.

## PAP & CHAP

- PAP is a two-way handshake where the client simply sends the password over to the server in the first LCP packet
- PAP sends the password in clear text, and there are no periodic checks after the connection establishes, meaning that a password change after establishment has no effect whatsoever
- CHAP is a three-way handshake where the server asks the client to authenticate (challenge), the client responds and the server accepts the connection
- CHAP sends the password (not the username) encrypted, there are also peroidic checks after the link establishes
- The default CHAP username is the hostname of the router

**R1 requests CHAP from R2:**

```
username R2 password cisco2
int se1/0
 encapsulation ppp
 ppp authentication chap
 ppp pap sent-username R1 password cisco1
```

**R2 requests PAP from R1:**

```
username R1 password cisco1
int se1/0
 encapsulation ppp
 ppp chap hostname R2
 ppp chap password cisco2
 ppp authentication pap

show users
who
debug ppp authentication
```

## AAA Authentication for PPP

- When using AAA the autocommand will only function if aaa authorization is also configured.
- Preferably use a private Radius or Tacacs+ server in combination with PPP authentication.

**R1 requests EAP from R2 and authenticates using Radius:**

```
aaa new-model
aaa group server radius MYRADIUS
 server-private 1.1.1.1 timeout 5 retransmit 0 key cisco

aaa authentication ppp PPP_R1_R2 group MYRADIUS local
aaa authorization exec default group MYRADIUS local
username R2 password cisco
username R2 autocommand logout

int se1/0
 encapsulation ppp
 ppp authentication eap PPP_R1_R2
 ppp eap local
 ppp chap hostname R1
 ppp chap password cisco

show aaa servers
show radius server-group all
debug radius
```

**R2 requests MS-CHAP-V2 from R2 and authenticates using Tacacs+:**

```
aaa new-model
aaa group server tacacs+ MYTACACS
 server-private 2.2.2.2 single-connection key cisco

aaa authentication ppp PPP_R1_R2 group MYTACACS local-case
aaa authorization exec default group MYTACACS local
username R1 password cisco
username R1 autocommand logout

int se1/0
 encapsulation ppp
 ppp eap identity R2
 ppp eap password cisco
 ppp authentication ms-chap-v2 PPP_R1_R2

show aaa servers
show tacacs private
debug tacacs
debug tacacs events
debug tacacs packets
```

## Multilink PPP (MLPPP)

- Uses a fragmentation scheme where large packets are sliced in pieces and sequence numbers are added using headers.

- Fragments are sent over multiple links and reassembled at the opposite end.
- Small voice packets are interleaved with fragments of large packets using a special priority queue.

The **interleave** keyword enables real-time packet interleaving.

- Allows large packets to be MLPPP encapsulated and fragmented into a small enough size to satisfy delay requirements.

```
int multilink 1
 ppp multilink interleave
 ppp multilink

int se1/0
 ppp multilink
 ppp multilink group 1
```

# IPCP

## Internet Protocol Control Protocol (IPCP)

- IPCP relies on PPP
- IOS ignores mask requests and offers. This is a problem when running RIP
- Use PPPoE with import IPCP into DHCP to acquire the correct mask
- Disable validate-update source in RIP when using IPCP

```
ip local pool IPCP 10.0.12.2
int se1/0
 ip add 10.0.12.1 255.255.255.0
 encapsulation ppp
 peer default ip address pool IPCP
 peer default ip address 10.0.12.2
 ppp ipcp mask 255.255.255.0
 peer neighbor-route

int se1/0
 encapsulation ppp
 ip address negotiated
 ppp ipcp mask request
 peer neighbor-route
 no shut

router rip
 no auto
 version 2
 network 10.0.12.0
 network 192.168.0.0
 no validate-update source
```

## Import IPCP subnet settings to local DHCP

- This will allow the import of the correct subnet mask
- The imported IPCP pool will always start at the first address (.1) even if only a single address is specified in the pool
- This will ignore additional IPCP settings, such as the default route installed through IPCP

```
ip dhcp pool LOCAL
 import all
 origin ipcp

int se1/0
 encapsulation ppp
 ppp ipcp mask request
 no ppp ipcp route default
 no ip add negotiated
 ip add pool LOCAL
```

# PPPoE

## PPP over Ethernet (PPPoE)

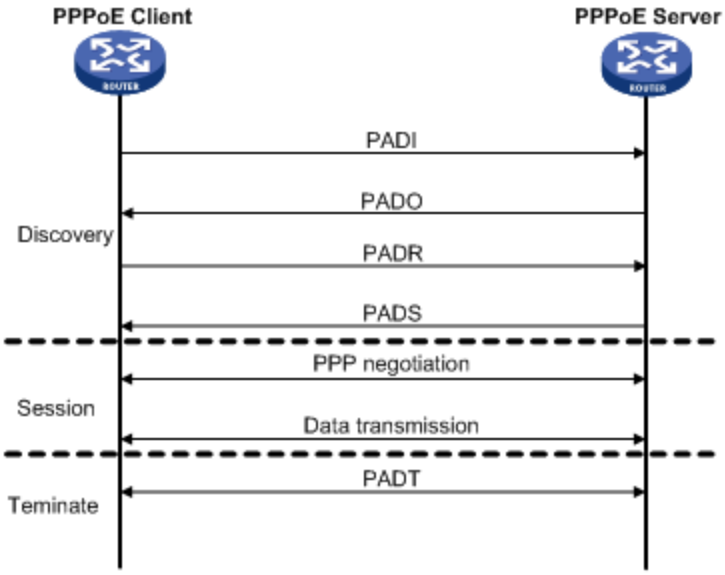
- PPPoE provides a standard method of employing the authentication methods of PPP over an Ethernet network
- Allows authenticated assignment of IP addresses
- The MTU size is automatically set to 1492 bytes

PPPoE consists of two phases:

- Active Discovery Phase. The client tries to locate the server (called an access concentrator in PPPoE terms)
  - During this phase the session ID is assigned and the PPPoE layer is established
- PPP Session Phase. PPP options are negotiated and authentication is performed
  - After session establishment, PPPoE functions as a L2 encapsulation method, using the PPP link with PPPoE headers

**Active Discovery Phase**

1. Client sends a broadcast PPPoE Active Discovery Initiation (PADI)
2. Server responds with unicast PPPoE Active Discovery Offer (PADO)
3. Client sends a unicast PPPoE Active Discovery Request (PADR)
4. Server allocates a unique session ID to the client and responds with a PPPoE Active Discovery Session (PADS)
5. PPPoE Active Discovery Termination (PADT) closes the connection when needed




**PPP Session Phase**

1. Link Control Protocol (LCP) initiation
2. Link establishment and authentication (PAP / CHAP)
3. Network Control Protocol (NCP) stage (IPCP or DHCP)

## PPPoE IPCP

- IOS ignores mask requests and offers. This is a problem when running RIP
- Use PPPoE with import IPCP into DHCP to acquire the correct mask

**Server:**

```
bba-group pppoe R2
 virtual-template 12
int fa0/0
 pppoe enable group R2

int lo0
 ip add 10.0.12.1 255.255.255.255

int virtual-template 12
 description R2
 ip address unnumbered lo0
 ip mtu 1492
 encapsulation ppp
 ppp authentication chap
 peer default ip address pool IPCP
 ppp ipcp mask 255.255.255.255

username R2 password cisco
ip local pool IPCP 10.0.12.2
```

**Client:**

```
int fa0/0
 pppoe-client dial-pool-number 12

interface Dialer 1
 ip address negotiated
 ip mtu 1492
 encapsulation ppp
 ppp chap hostname R2
 ppp chap password cisco
 dialer pool 12
 ppp ipcp mask request
 ppp ipcp route default
```

## PPPoE IPCP with local DHCP

- Import IPCP subnet settings to local DHCP. Will allow the import of the correct subnet mask
- The imported IPCP pool will always start at the default first address (.1) even if only a single address is specified in a pool
- This will ignore additional IPCP settings, such as the default route installed through IPCP
- Requires bouncing of the interfaces if overwriting an existing IPCP configuration

**Client:**

```
ip dhcp pool IMPORT_IPCP
 import all
 origin ipcp

int dialer1
 dialer pool 12
 ip mtu 1492
 encapsulation ppp
 ppp chap username R2
 ppp chap password cisco
 ppp ipcp mask request
 ip address pool IMPORT_IPCP

int fa0/0
 pppoe-client dial-pool-number 12
```

## PPPoE DHCP

- Use DHCP reservation based on client-identifier
- Stay away from DHCP excluded address ranges

**Server:**

```
bba-group pppoe R2
 virtual-template 12
```

```
int fa0/1
 pppoe enable group R2

interface Virtual-Template12
 description R2
 ip address 10.0.12.1 255.255.255.252
 ip mtu 1492
 peer default ip address dhcp-pool DHCP_HOST
 ppp authentication pap

username R2 password cisco
ip dhcp pool DHCP_HOST
 host 10.0.12.2 /30
 client-identifier 0063.6973.636f.2d63.6130.332e.3065.3434.2e30.3030.382d.4661.302f.30
```

**Client:**

```
int fa0/1
  pppoe-client dial-pool-number 12

interface Dialer 1
 dialer pool 12
 ip mtu 1492
 encapsulation ppp
 ppp pap sent-username R2 password cisco
 ip address dhcp
```

# PBR

## Policy-Based Routing (PBR)

- PBR intercepts a (matched) packet before it is sent to the RIB (or CEF) and applies its policies
- On older devices PBR traffic was only process switched, on newer devices PBR is cef switched
    - Enable fast switching of PBR traffic on older devices with **ip route-cache policy** on the interface
- If the packet matches the ACL, the RIB is not consulted unless the outgoing interface that PBR uses is down
    - This is only applicable if the local router interface is down (admin down or protocol down)
    - If the opposite side is down, then the PBR will black hole traffic (unless you implement IP SLA as well, see below)
- A **permit** clause in the route-map tells IOS to PBR route the packet, a **deny** clause will forward the packet to the regular routing engine

- Only the first statement (10) is matched in route-maps if there is a match
    - If there is no match in the first statement, traffic is routed normally
    - For this reason PBR route-maps allow more than one **set** statement in route-map sequence

## PBR Route-Map Statements

- The **set ip next-hop** and **set interface** are used unconditionally unless the outgoing interface is down
- The **set ip default next-hop** and **set default interface** apply to the default route and are used before the regular default route.
- The idea is to specify an alternate default route for hosts matched in the ACL

The logical difference between default and non-default statement:

| **Statement** | **Logical action** |
|---|---|
| non-default | PBR first, if no match use RIB<br>If outgoing interface down use RIB |
| default | RIB first with exception of default route, PBR default route second, RIB default route third |

```
ip access-list standard PBR_ACL
 permit 172.16.0.0 0.0.0.255

route-map PBR permit 10
 match ip address PBR_ACL
 set ip default next-hop 10.0.12.2
 set default interface se1/0

interface fa0/0
 ip policy route-map PBR

show ip policy
debug ip policy
```

There's an order of operations to PBR set statements. ip next-hop -> interface -> ip default next-hop -> default interface

- If the first statement fails the next will be evaluated. Remember that addresses are preferred over interfaces
- The **recursive** keyword can be used to specify a next-hop that is not directly connected

## Local PBR

- Applying the PBR on an interface does not affect traffic locally generated by the router (even when sourcing the interface)
- Create an **ip local policy** using loopbacks to policy-route local traffic
- Use extended access-lists to have more granular control over which local traffic is policy routed
- Using standard access-lists will policy route all local traffic

```
ip access-list standard PBR_LOCAL_ACL
 permit host 192.168.0.1
```

```
route-map PBR_LOCAL permit 10
 match ip address PBR_LOCAL_ACL
 set ip next-hop 10.0.12.2
 set interface se1/0
ip local policy route-map PBR_LOCAL

show ip local policy
```

## PBR, IP SLA & Tracking

- Combine tracking object, IP SLA object and PBR route-map to use PBR depending on reachability
- Use the **verify-availability** in the route-map statement
- Format **ip next-hop verify-availability <remote-ip> <sequence number> track <track object>**

```
ip sla 1
 icmp-echo 10.0.12.2 source-ip 10.0.12.1

ip sla schedule 1 start now life forever
track 1 ip sla 1

route-map PBR permit 10
 match ip address 1
 set ip next-hop verify-availability 10.0.12.2 1 track 1

int gi0/0
 ip policy route-map PBR
```

- When tracking object 1 goes from UP > DOWN, the PBR next-hop clause will be deactivated and traffic is routed normally

# SNMP

## Simple Network Management Protocol (SNMP)

| Terminology | Description | UDP port | Configured with |
|---|---|---|---|
| SNMP Manager<br>NMS | Server that runs the SNMP polling software<br>Also called Network Management Server (NMS) | - | - |
| SNMP Agent | Device that sends SNMP info to NMS | - | - |
| MIB | Management Information Base (MIB)<br>Structure of objects holding information about the device | - | - |

| | | | |
|---|---|---|---|
| SNMP GET | A NMS directly requests information from an SNMP Agent (read-only)<br>NMS can only poll on specific intervals, so information might be lost | 161 | snmp-server community … ro |
| SNMP SET | Allows an NMS to make changes to the device configuration (read/write) | 161 | snmp-server community … rw |
| SNMP TRAP (default) | Unacknowledged sent from Agent to NMS<br>Traps are sent when an event occurs and not on a polling interval | 162 | snmp-server host …. traps |
| SNMP INFORM | Acknowledged sent from Agent to NMS<br>Informs are sent when an event occurs and not on a polling interval | 162 | snmp-server host …. informs |

SNMP Versions

## SNMP Communities / Users & Groups

- Both v1 and v2 groups are created when configuring a SNMP community
  - The default read view is v1default
  - The default write view is v1default
- Disable the v1 group with the **no snmp-server group public v1** command
- Also disable the Interim Local Management Interface (ILMI) SNMP groups
  - The ILMI community itself cannot be deleted

```
snmp-server community public ro
no snmp-server group public v1
no snmp-server group ILMI v1
no snmp-server group ILMI v2c
```

```
show snmp community
Community name: ILMI
Community Index: cisco0
Community SecurityName: ILMI
storage-type: read-only  active

Community name: public
Community Index: cisco7
Community SecurityName: public
storage-type: nonvolatile       active

show snmp group
groupname: public                          security model:v1
contextname: <no context specified>        storage-type: permanent
```

```
readview : v1default                          writeview: <no writeview specified>
notifyview: <no notifyview specified>
row status: active

groupname: public                             security model:v2c
contextname: <no context specified>           storage-type: permanent
readview : v1default                          writeview: <no writeview specified>
notifyview: <no notifyview specified>
row status: active
```

- SNMP community is basically a combination of a username and password
- SNMP users and groups are not only a v3 concept, a combination of the two is basically the same as a community
- It is possible to configure SNMP users/groups for v1 and v2c

```
snmp-server user publicwrite publicwrite v2c
snmp-server group publicwrite read v1default write v1default
```

```
show snmp community
Community name: publicwrite
Community Index: cisco5
Community SecurityName: publicwrite
storage-type: nonvolatile       active

show snmp group
groupname: publicwrite                        security model:v2c
contextname: <no context specified>           storage-type: nonvolatile
readview : v1default                          writeview: v1default
notifyview: <no notifyview specified>
row status: active
```

## SNMP Host

- Only SNMP Traps will be sent to the host, unless you specify the **inform** keyword
- SNMP v1 is the default when not specifying a version

```
snmp-server enable traps
snmp-server host 1.1.1.1 traps version 2c public udp-port 162
snmp-server host 1.1.1.1 inform version 2c public udp-port 162
no snmp-server group public v1
```

```
snow snmp host
Notification host: 1.1.1.1      udp-port: 162   type: inform
user: public    security model: v2c
```

## SNMPv3

- The SNMP group security level is a minimum allowed security level
- The actual security level for the user is defined in the **snmp-server** user command.
  - This is the minimum security level for that specific user
- Other users may still connect using the minimum allowed group security level

| Security level | Authentication method | Encryption support |
|---|---|---|
| SNMPv1 | Community string | - |
| SNMPv2c | Community string | - |
| noAuthNoPriv | Username Only | - |
| authNoPriv | MD5 or SHA-1 | - |
| authPriv | MD5 or SHA-1 | DES, 3DES or AES |

```
snmp-server group GROUP v3 priv
snmp-server user USER GROUP v3 auth sha cisco priv aes 256 cisco
snmp-server host 1.1.1.1 traps version 3 priv USER
```

```
show snmp user
User name: USER
Engine ID: 800000090300CA0112100006
storage-type: nonvolatile       active
Authentication Protocol: SHA
Privacy Protocol: AES256
Group-name: GROUP

show snmp group
groupname: GROUP                          security model:v3 priv
contextname: <no context specified>       storage-type: nonvolatile
readview : v1default                      writeview: <no writeview specified>
notifyview: *tv.FFFFFFFF.FFFFFFFF.FFFFFFFF.F
row status: active

show snmp host
Notification host: 1.1.1.1      udp-port: 162   type: trap
user: USER      security model: v3 priv
```

## SNMP Filtering

```
ip access-list standard SNMP
 permit host 1.1.1.1
 deny any log
```

```
snmp-server community public ro SNMP
```

### SNMPv3 Filtering

```
ip access-list standard SNMP_USER
 permit host 1.1.1.1
ip access-list standard SNMP_GROUP
 permit host 1.1.1.1

snmp-server user USER GROUP v3 auth sha cisco priv aes 256 cisco access SNMP_USER
snmp-server group GROUP v3 priv access SNMP_GROUP
```

### SNMP Engine-ID

- SNMPv3 user passwords are hashed based on the value of the local Engine-ID
- If the Engine-ID changes, the security digests of SNMPv3 users will be invalid, and the users will have to be reconfigured
- Trailing zeroes will be added automatically to create 24 characters when changing the Engine-ID

```
snmp-server engine-id local 123412341234
snmp-server engine-id remote 192.168.0.1 123412341234
```

# System

### Memory Reservations

```
memory free low-watermark processor
memory reserve critical
memory reserve console

show memory console reserved
```

### CPU Threshold

- The rising and falling commands trigger a syslog message when CPU is above/below threshold.

```
snmp-server enable traps cpu threshold
process cpu threshold type total rising 50 interval 5 falling 10 interval 5
```

# TCL

**TCL Scripting**

```
tclsh
```

```
foreach X {
192.168.0.1
192.168.0.2
192.168.0.3
192.168.0.4
192.168.0.5
192.168.0.6
192.168.0.7
192.168.0.8
192.168.0.9
192.168.0.10
2001:192:168::1
2001:192:168::2
2001:192:168::3
2001:192:168::4
2001:192:168::5
2001:192:168::6
2001:192:168::7
2001:192:168::8
2001:192:168::9
2001:192:168::10
} { ping $X repeat 100 time 1 source loopback 0}
```

# Traceroute / Ping

## Traceroute

- Traceroute has default TTL of 30.
- Steps performed when traceroute is executed:
    1. Sends 3 packets with TTL=1 to first-hop router. FH router responds with time-exceeded (ICMP Type-11).
    2. In response sends 3 packets with TTL=2 to FH router, second-hop router responds with TTL message.
    3. Continues until packets arrive at destination, last-hop router responds with (ICMP Type-3).
    4. The LH router sends back a unreachable message because the destination is an unreachable port.

Traceroute Output

The * means that ICMP rate limit is enabled at the last-hop router. The default timeout is 500 msec.

- The reason only the LH router shows this is because intermediate routers send a time exceeded TTL message.
- The second traceroute packet usually times out because that one is within the 500 msec interval, the third packet is not.
- The same applies to ping with U.U.U output, the 1st message is sent back as unreachable by the LH router. The 2nd times out because it is within the 500 msec interval, the 3rd is unreachable again and so on...

## TTL

- Time to live values are different between vendors / operating systems
  - Linux distributions (including macOS) often have a default TTL value of 64
  - Windows usually sets the default TTL value to 128
  - Junos devices have a default value of 64
  - Cisco devices have  a default value of 255

```
ip icmp rate-limit unreachable 500

show ip icmp rate-limit
```

## Traceroute Responses

| | |
|---|---|
| * | The probe timed out |
| A | Administratively prohibited (ACL) |
| U | Port unreachable |
| H | Host unreachable |
| N | Network unreachable |

## Ping

- Time exceeded on ping means TTL expired (ICMP Type-11). This is used in traceroute
- Specify how often ICMP unreachable messages are sent to neighbors with the **ip icmp rate-limit unreachable** command (500 default)
- ICMP redirect messages are used to notify hosts that a better route (other router) is available for a particular destination
- The kernel is configured to send redirects by default. Disable with the interface command **no ip redirects**

Cisco routers send ICMP redirects when all of these conditions are met:

- The ingress interface is the same as the egress interface of the packet
- The source is on the same subnet as the better next-hop
- The source does not use source-routing

## ICMP types

| | |
|---|---|
| 0 | Echo Reply |
| 3 | Destination Unreachable |
| 5 | Redirect |
| 8 | Echo |
| 11 | Time Exceeded (TTL) |

## ICMP Responses

| | |
|---|---|
| ! | Reply |
| . | Timed Out |
| U | Destination Unreachable |

# Tracking

## Object Tracking

- Interface ip / ip routing - Track the presence of an ip address
- Line-protocol - Track the line-protocol (up/down)
- IP route - Track a route present in the routing table
- IP SLA - Track an IP SLA object
- List - Combine multiple track objects based on percentage/weight or boolean AND/OR

## Tracking Options

- State - Up or down (default)
- Reachability - Up or down + within configured threshold (timeout settings)

```
track 1 int gi0/0 ip routing
track 2 int gi0/1 line-protocol
track 3 ip sla 1 state
track 4 ip route 0.0.0.0/0 reachability
```

**Set a tracking delay on objects to delay the transitioning of states**

```
track 3 ip sla 1
 delay up 15 down 25
```

- If the tracking object is UP, after ip sla 1 times out, the tracking object will wait 25 seconds before transitioning to a DOWN state
- If the tracking object is DOWN, after ip sla 1 regains connectivity, the tracking object will wait 15 seconds before transitioning to an UP state

```
show track
Track 1
  Interface GigabitEthernet0/0 ip routing
  IP routing is Up
    1 change, last change 00:01:18
Track 2
  Interface GigabitEthernet0/1 line-protocol
```

```
  Line protocol is Up
    1 change, last change 00:01:18
Track 3
  IP SLA 1 state
  State is Up
    1 change, last change 00:01:18
  Delay up 15 secs, down 25 secs
  Latest operation return code: OK
  Latest RTT (millisecs) 7
Track 4
  IP route 0.0.0.0 0.0.0.0 reachability
  Reachability is Up (static)
    2 changes, last change 00:00:02
  First-hop interface is GigabitEthernet0/0
```

**show track brief**

| Track | Object | | Parameter | Value | Last Change |
|---|---|---|---|---|---|
| 1 | interface | GigabitEthernet0/0 | ip routing | Up | 00:03:09 |
| 2 | interface | GigabitEthernet1/0 | line-protocol | Up | 00:03:09 |
| 3 | ip sla | 1 | state | Up | 00:03:09 |
| 4 | ip route | 0.0.0.0/0 | reachability | Up | 00:01:53 |

## IP route metric threshold

- Whatever the metric is will be given a value from 1-255 based on the track resolution
- If route metric is under the up threshold, tracking is up
- Optionally change resolution of specific routing protocol

```
track 5 ip route 0.0.0.0/0 metric threshold
 metric threshold up 50

track resolution ip route eigrp 2560
show track resolution
```

## Combine Track Objects

Boolean tracking:

- If both objects up, track 12 is up
- If either object is down, track 12 is down

```
track 12 list boolean and
 object 1
 object 2
```

## Percentage tracking

- If two out of three objects are up(66%), track 123 is up
- If one out of three objects is up (33%), track 123 is down

- Track 123 will stay down until two out of tree objects are back up (66%)

```
track 123 list threshold percentage
 object 1
 object 2
 object 3
 threshold percentage up 66 down 33
```

### Weighted tracking

- If object 2 is down, track 234 is down
- If either object 3 or 4 is down, track 234 is up

```
track 234 list threshold weight
 object 2 weight 100
 object 3 weight 50
 object 4 weight 50
 threshold weight up 150 down 100
```

### Track Timers

- Not every object type is tracked at the same rate
- Manually change track timers to allow for faster detection

```
track timer ip route 5
show track timers
```

# IP Routing

### Administrative Distance and Route Selection

- Hardcoded original administrative distance will win if different routing protocols are configured to use the same AD.
- The metrics between different routing protocols or different routing processes are not compared in the route selection.
- When receiving the same route from different OSPF processes with the same AD, the route learned first wins.
- OSPF does not differentiate between internal and external routes to the same destination. Only the AD matters.
- When receiving the same route from different EIGRP ASs with the same AD, the route from the lower AS wins.

### IP Source-Routing

- Allows the originator of a packet to decide which routers the packet will flow through.

- Basically a custom path of all hops specified at the source and set in the actual IP header by the source.
- Enabled by default but is a security risk. Disable with no ip source-route command.

## Bandwidth Delay Product (BDP)

- Maximum number of bits that can be on a network segment at the same time
- Calculated using the bandwidth (in bits/sec) and the latency (in sec)
- Latency can (depending on the exam question) be either:
    - Round-trip time (RTT) between source and destination nodes
    - Unidirectional delay from one node to another

**64kbps link with 3 second RTT**
64000 bits/sec * 3 = 192000 bits = 24000 bytes

## IP Accounting

- Counts the number of IP packets and logs source/destination.
- Only works for transit egress traffic, not local traffic.

```
int fa0/0
 ip accounting output-packets

show ip accounting
```

## IP Redirects

- ICMP redirect messages are send by default when routers have to forward a packet on the same interface it was received.
- Routers will notify hosts of better next-hop through redirects.

```
int fa0/0
 no ip redirects
 no ipv6 redirects
```

## IP Unreachables

- By default the router will respond with an IP unreachable ICMP message in case the neighbor router pings an unknown address
- Disable to block UDP port scans. These have a destination of an unused or unreachable UDP port
- IP packets for unknown destinations (in case of EIGRP discard route for example) are sent to null0
    - Disable redirects for all discarded unknown traffic directly on the null0 interface instead of each separate interface
    - Other interfaces will still respond with unreachable if traffic is destined to the local interface address

```
int fa0/0
```

```
no ip unreachables
no ipv6 unreachables

int null0
 no ip unreachables
 no ipv6 unreachables
```

## IP Local Proxy-ARP

- Enable proxy of ARP request on the same subnet with the ip local proxy-arp interface command.
- The ip proxy-arp feature is for ARP requests to different subnets. Enabled by default.
- Use in Private-VLANs to allow communication between isolated hosts. Configure on the promiscuous port.
- Instead of configuring local proxy-arp, you can also statically configured the IP to MAC mappings for individual hosts.

```
arp 10.0.123.2 abcd.1234.abcd arpa
arp 10.0.123.3 1234.abcd.1234 arpa
```

## IP Directed-Broadcast (SMURF)

- Disabled by default. Exploited in SMURF attacks.
- Enable with ip directed-broadcasts command on interfaces.

## IP Event Dampening

- Suppress flapping interface effects on routing protocols and routing tables.
- Can only be configured on physical interfaces, not on sub-interfaces or virtual-templates.

```
int fa0/0
 dampening 5 1000 2000 20

show interfaces dampening
```

# ODR

## On-Demand Routing (ODR)

- Disable CDP on outside (physical) interfaces. Enable CDP on tunnel interfaces.
- ODR timers and CDP timers should be the same.
- Configure on hub only, spokes will receive a default route via ODR and will advertise their connected subnets.
- In P2 all traffic will still flow through the hub, ODR needs P3 in order to create dynamic tunnels.

# Redistribution

## Redistribution

- Redistribution only redistributes routes that are present in the RIB
  - There is no direct redistribution between protocols
- Routing protocol redistribution also redistributes the connected networks that the protocol is enabled for
- You can either set route-tags with a route-map or directly in the redistribution statement
  - Filtering can then be done through **distribute-lists** matching route-maps
- Include IGP interfaces when filtering redistributed connected routes (loopbacks)
  - Another way to include the connected interfaces is to advertise them into the protocols and optionally configure as passive

- OSPF default static route cannot be redistributed with the **redistribute static** command, even if a route-map is specified
  - Always redistribute the default route into OSPF using the **default-information originate command**.

## BGP Redistribution Rules

- Only internal OSPF routes will be redistributed into BGP by default
- A default route learned by an IGP is not redistributed into BGP by default
- Advertise the redistributed default route with **network 0.0.0.0 mask 0.0.0.0** statement
- The route-tag associated with the redistributed route is the neighbor AS, not the OSPF PID or EIGRP ASN etc.
- If a metric is given to a route, it will become the BGP MED

```
router bgp 12
 address-family ipv4
  redistribute ospf 1 metric 50 match internal
```

## OSPF Redistribution Rules

- Routes redistributed from BGP into OSPF will receive a cost of 1
- Routes redistributed from OSPF into OSPF will retain their **original** cost values
- Routes redistributed from anything else into OSPF will receive a cost of 20 (E2)
- Default external metric is 20 with a metric-type of 2 (E2)
- Redistributes only classful (Class A, B, and C) by default, use **subnets** keyword

## BGP into OSPF Defaults

```
redistribute bgp <ASN> subnets
```

**Will show up in the running config as:**

```
redistribute bgp <ASN> metric 20 metric-type 2 subnets tag 0
```

- **Tag 0** means that the tag will not be changed, by default when you redistribute the routes will receive the tag associated with the ASN
    - Redistributed routes from BGP 64512 into OSPF will receive the tag 64512
- Even though the metric says 20 in the output above, the cost associated with the route will be 1 unless you hardcode it
    - So when you hardcode configure **redistribute bgp <ASN> subnets metric 20**, the cost of the routes will be 20

## OSPF into OSPF Defaults

```
redistribute ospf <PID> metric 20 metric-type 2 subnets tag 0 match internal external 1 external 2 nssa-external 1 nssa-external 2
```

- Tag 0 means that the tag will not be changed, by default when you redistribute the routes will not receive a tag
- Even though the metric says 20 in the output above, the routes will retain their original costs
    - If you change the metric-type to E1, the costs will be incremented (1 in the case of Fast- or GigabitEthernet) by the redistributing router
    - If you leave the metric-type as E2 (default) or hardcode it, the costs will be the original value before redistribution (without addition)

**Example #1**

```
router ospf 12
 redistribute ospf 1 subnets metric-type 1 metric 5 match external 1
```

- Only OSPF external type 1 (E1) routes in PID 1 will be redistributed into PID12
- The ASBR will rewrite whatever costs the routes had to 5 and will set the metric-type to E1
- The neighbor receiving the redistributed routes will see a cost value of 5 + its own interface cost towards the ASBR

**Example #2**

```
ip prefix-list PID1_PREFIX permit 1.1.1.1/32

route-map OSPF_TO_OSPF permit 10
 match ip address prefix-list PID1_PREFIX
 set metric-type type-1
 set metric 5
route-map OSPF_TO_OSPF permit 99

router ospf 12
 redistribute ospf 1 subnets match external 2 route-map OSPF_TO_OSPF
```

- Only OSPF external type 2 (E2) routes in PID 1 will be redistributed into PID12
- The ASBR will set the metric-type to E1 and to cost to 5 for the prefix 1.1.1.1/32 via a route-map
- All other prefixes will receive the metric-type E2 because of default configuration

- The **set metric** statement is smart enough to determine which metric it should set, the same command is used for BGP, RIP and EIGRP

## OSPF into EIGRP Defaults

- Routes from any protocol except static and connected need to be associated with a metric
- Either set the metric on each redistribution statement or configure a **default-metric**

```
router eigrp 12
 default-metric 1000000 1 255 1 1500
 redistribute ospf 1
```

- Redistributed routes will become EIGRP external routes with AD 170
- In the EIGRP topology the redistributing protocol will be listed as **'External Protocol'**
- The external metric is also displayed before redistribution and conversion to the EIGRP metric happened

```
R2#show ip eigrp topology 1.1.1.1/32
EIGRP-IPv4 Topology Entry for AS(12)/ID(0.0.0.2) for 1.1.1.1/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 3072
  Descriptor Blocks:
  10.0.12.1 (GigabitEthernet0/0), from 10.0.12.1, Send flag is 0x0
     Composite metric is (3072/2816), route is External
     Vector metric:
       Minimum bandwidth is 1000000 Kbit
       Total delay is 20 microseconds
       Reliability is 255/255
       Load is 1/255
       Minimum MTU is 1500
       Hop count is 1
       Originating router is 0.0.0.1
     External data:
       AS number of route is 1
       External protocol is OSPF, external metric is 20
       Administrator tag is 0 (0x00000000)
```

## Redistribution Scenario

- Configure an **ip route profile** to test the data plane for routing loops
- Enable **debug ip routing** to test the control plane for routing loops

```
ip route profile
show ip route profile

debug ip routing
```

**Redistribution using Direct Tags**

```
route-map EIGRP_ROUTES deny 10
 match tag 90
route-map EIGRP_ROUTES permit 99

router ospf 1
 redistribute eigrp 1 subnets tag 90
  distribute-list route-map EIGRP_ROUTES in
```

**Mutual MultiPoint Redistribution using Prefix-Lists**

```
ip prefix-list OSPF_ROUTES permit 3.0.0.1/32
ip prefix-list OSPF_ROUTES permit 3.0.0.2/32
ip prefix-list OSPF_ROUTES permit 3.0.0.3/32

ip prefix-list EIGRP_ROUTES permit 4.0.0.1/32
ip prefix-list EIGRP_ROUTES permit 4.0.0.2/32
ip prefix-list EIGRP_ROUTES permit 4.0.0.3/32

route-map EIGRP_TO_OSPF deny 10
 match ip address prefix-list OSPF_ROUTES
route-map EIGRP_TO_OSPF permit 20
 match ip address prefix-list EIGRP_ROUTES

route-map OSPF_TO_EIGRP deny 10
  match ip address prefix-list EIGRP_ROUTES
route-map OSPF_TO_EIGRP permit 20
 match ip address prefix-list OSPF_ROUTES

router eigrp 1
 redistribute ospf 1 metric 1000000 10 255 1 1500 route-map OSPF_TO_EIGRP

router ospf 1
 redistribute eigrp 1 metric-type 1 subnets route-map EIGRP_TO_OSPF
```

**Three-Way Redistribution using Tags**

```
route-map EIGRP_TO_RIP deny 10
 match tag 120
route-map EIGRP_TO_RIP permit 20
 match tag 110
 set tag 110
route-map EIGRP_TO_RIP permit 30
 set tag 90

route-map RIP_TO_EIGRP deny 10
 match tag 90
route-map RIP_TO_EIGRP permit 20
 match tag 110
```

```
 set tag 110
route-map RIP_TO_EIGRP permit 30
 set tag 120

route-map OSPF_TO_RIP deny 10
 match tag 120
route-map OSPF_TO_RIP permit 20
 match tag 90
 set tag 90
route-map OSPF_TO_RIP permit 30
 set tag 110

route-map RIP_TO_OSPF deny 10
 match tag 110
route-map RIP_TO_OSPF permit 20
 match tag 90
 set tag 90
route-map RIP_TO_OSPF permit 30
 set tag 120

route-map OSPF_TO_EIGRP deny 10
 match tag 90
route-map OSPF_TO_EIGRP permit 20
 match tag 120
 set tag 120
route-map OSPF_TO_EIGRP permit 30
 set tag 110

route-map EIGRP_TO_OSPF deny 10
 match tag 110
route-map EIGRP_TO_OSPF permit 20
 match tag 120
 set tag 120
route-map EIGRP_TO_OSPF permit 30
 set tag 90

router eigrp 1
 redistribute ospf 1 metric 1000000 1 255 1 1500 route-map OSPF_TO_EIGRP
 redistribute rip metric 1000000 1 255 1 1500 route-map RIP_TO_EIGRP

router ospf 1
 redistribute eigrp 1 metric-type 1 subnets route-map EIGRP_TO_OSPF
 redistribute rip metric-type 1 subnets route-map RIP_TO_OSPF

router rip
 redistribute eigrp 1 metric 3 route-map EIGRP_TO_RIP
 redistribute ospf 1 metric 3 route-map OSPF_TO_RIP
```

# VRF-Lite

## VRF-Lite

- Divide interfaces into VRFs and create separate routing tables
- Do not implement L3VPN afterwards, this is why its called VRF-Lite

```
vrf definition 10
 add ipv4
vrf definition 172
 add ipv4
int fa0/0
 vrf forwarding 10
 ip add 10.0.12.1 255.255.255.0
int fa0/1
 vrf forwarding 10
 ip add 10.0.13.1 255.255.255.0
int se1/0
 vrf forwarding 172
 ip add 172.0.12.1 255.255.255.0
int se1/1
 vrf forwarding 172
 ip add 172.0.13.1 255.255.255.0

router eigrp EIGRP
 address-family ipv4 unicast vrf 10 autonomous-system 10
  no auto-summary
  network 10.0.12.0 0.0.0.255
  network 10.0.13.0 0.0.0.255
address-family ipv4 unicast vrf 172 autonomous-system 172
  no auto-summary
  network 172.0.12.0 0.0.0.255
  network 172.0.13.0 0.0.0.255
```

**Non-VRF neighbors**

```
router eigrp 10
  no auto-summary
  network 10.0.12.0 0.0.0.255
router eigrp 172
  no auto-summary
  network 172.0.12.0 0.0.0.255
```

# EVN

## Easy Virtual Networking

- Extension of VRF-Lite
- Interfaces acting as trunks between routers are divided into sub-interfaces when using VRF-Lite
  - Uses 802.1Q encapsulation and subinterfaces to differentiate between VRFs
  - This method can lead to many subinterfaces
- The solution is to use EVN and create 'VNET trunks'
- The concept is similar to regular trunks and VLANs, in this case the VLANs are the VRFs
- Only supports static routes, OSPFv2 and EIGRP for unicast routing

**Advantages (according to Cisco)**

- Simplified Layer 3 network virtualization
- Improved shared services support
- Enhanced management, troubleshooting and usability

## VRF trunks config without EVN

**R1**

```
vrf definition A
 vnet tag 10
 address-family ipv4
vrf definition B
 vnet tag 172
 address-family ipv4

int gi0/1
 vrf forwarding A
 ip add 10.0.10.1 255.255.255.0
int gi0/2
 vrf forwarding B
 ip add 172.16.10.1 255.255.255.0

int gi0/0
 description TO_R2
int gi0/0.10
 description VRF_A
 encapsulation dot1Q 10
 vrf forwarding A
 ip address 10.0.12.1 255.255.255.252
int gi0/0.20
 description VRF_B
 encapsulation dot1Q 10
 vrf forwarding B
 ip address 10.0.12.1 255.255.255.252
```

## VRF trunks config with EVN

- The VNET tag identifies the VRF and associates it with a dot1q tag (basically a VLAN-id)

- Each VRF added will automatically added to the VNET Trunk
- EVN basically creates a sub-interface for each interface and associates a 'vrf forwarding' instance

**R1**

```
vrf definition A
 vnet tag 10
 address-family ipv4
vrf definition B
 vnet tag 172
 address-family ipv4

int gi0/1
 vrf forwarding A
 ip add 10.0.10.1 255.255.255.0
int gi0/2
 vrf forwarding B
 ip add 172.16.10.1 255.255.255.0

int gi0/0
 description TO_R2
 vnet trunk
 ip add 10.0.12.1 255.255.255.252
```

**R2**

```
vrf definition A
 vnet tag 10
 address-family ipv4
vrf definition B
 vnet tag 172
 address-family ipv4

int gi0/1
 vrf forwarding A
 ip add 10.0.20.2 255.255.255.0
int gi0/2
 vrf forwarding B
 ip add 172.16.20.2 255.255.255.0

int gi0/0
 description TO_R1
 vnet trunk
 ip add 10.0.12.2 255.255.255.252
```

**Verify**

show vrf

**R1**

```
ping vrf A 10.0.10.100
ping vrf B 172.0.10.100
```

**R2**

```
ping vrf A 10.0.20.100
ping vrf B 172.0.20.100
```

### Routing Between VRFs

- In the above example there is no reachability between hosts behind R1 vrf A and R2 vrf A
- This is because they are not on the same network (10.0.10.0/24 and 10.0.20.0/24)
- Optionally add routing protocols to provide reachability between these VRFs if they are on different networks

**R1**

```
router ospf 1 vrf A
 network 10.0.10.0 0.0.0.255 area 0
 network 10.0.12.0 0.0.0.3 area 0
router ospf 1 vrf B
 network 172.0.10.0 0.0.0.255 area 0
 network 10.0.12.0 0.0.0.3 area 0
```

**R2**

```
router ospf 1 vrf A
 network 10.0.20.0 0.0.0.255 area 0
 network 10.0.12.0 0.0.0.3 area 0
router ospf 1 vrf B
 network 172.0.20.0 0.0.0.255 area 0
 network 10.0.12.0 0.0.0.3 area 0
```

# IP Services

### Secure Copy Protocol (SCP)

- Requires SSH and AAA Authorization
- Has to be enabled on both routers to allow mutual copying

```
aaa new-model
aaa authentication login default local
aaa authorization exec default local
username bpin privilege 15 password cisco
ip scp server enable
```

```
copy scp://bpin@10.0.12.1/nvram:startup-config null:
```

## RCMD Remote-Copy (RCP) and Remote-Shell (RSH)

- Allow remote users to copy files to router with RCP.
- Allow remote users to execute commands with RSH.
- Server side has two names in the rcmd command.
    - First one must match /user on client.
    - Second one must match client hostname or client remote-username command.
    - The enable keyword allows execution of exec commands.

**Server**

```
ip rcmd rcp-enable
ip rcmd rsh-enable
ip rcmd remote-host remoteadmin 10.0.12.2 R2 enable

Configure the client:
ip rcmd remote-username R2

rsh 10.0.12.1 /user remoteadmin show ip interface brief
copy rcp://remoteadmin@10.0.12.1/nvram:startup-config null:
```

The boot/service config enables auto-loading of configuration files from a network server:

```
service config
ip rcmd remote-username R2
boot network rcp://10.0.12.1/BOOT
boot network tftp:BOOT
```

Local TFTP-Server

- Specify all files that are eligible for TFTP transfer separately.
- Optionally create an alias for the file and limit access.

```
access-list 1 permit host 192.168.0.2
tftp-server nvram:startup-config alias STARTUP 1

Configure client:
ip tftp source-interface loopback0
copy tftp://10.0.12.1/STARTUP null:
```

DNS Services

- Create individual host entries on the DNS server.

```
Server:
ip domain-lookup
ip domain-name lab.local
ip dns server
```

ip host Server1 2.2.2.2
ipv6 host Server2 2::2

Client:
ip domain-lookup
ip name-server 1.1.1.1
ip name-server 1::1

Configuration Generation Performance Enhancement (Parser)

- Caches interface configuration in memory, thus allowing faster execution of show run, write memory, etc.
- Enable with the parser config cache interface command.

TCP small servers

- Echo (7): Echoes back whatever you type.
- Chargen (19): Generates a stream of ASCII data.
- Discard (9): Throws away whatever you type.
- Daytime (13): Returns system date and time.

telnet 10.0.12.1 19

Terminate on server with:
show tcp brief
clear tcp tcb <tcb-value>

UDP small servers:

- Echo (7): Echoes the payload of the datagram you send.
- Discard (9): Silently pitches the datagram you send.
- Chargen (19): Pitches the datagram you send, and generates a stream of ASCII data.

Misc. Services
The X28 editor is enabled by default:
no service pad

Ensure that abnormally terminated TCP sessions are removed:
service tcp-keepalives-in
service tcp-keepalives-out

The finger service (TCP port 79) gives line information and is disabled by default:
ip finger
service finger

# DHCPv4

## DHCPv4 Messages

| Message | Type | Description |
|---|---|---|
| DHCPDISCOVER | Broadcast | Client tries to locate a server |
| DHCPOFFER | Broadcast<br>or Unicast | Server responds with DHCP options and proposed IP address<br>Sent using broadcast by default |
| DHCPREQUEST | Broadcast | Client accepts IP address assignment |
| DHCPACK | Broadcast<br>or Unicast | Acknowledgement by the DHCP server<br>Sent using broadcast by default |

The DHCPOFFER and DHCPACK are sent broadcast by default (on IOS devices)

- Request unicast from server with the **no ip dhcp-client broadcast-flag** global command (on the client)

Disable DHCP

```
no service dhcp
ip dhcp bootp ignore
```

**bootp ignore** - do not reply to Bootstrap Protocol request packets received

## DHCPv4 Conflict Logging

Similar function to excluded-addresses. Logs conflicts with a syslog message and stores the address on an exclusion list

- Conflicted addresses are stored and need clearing or restart to become usable again
- Enabled by default, disable with the **no ip dhcp conflict logging** command

## Common DHCPv4 Options

| Option # | Option Name |
|---|---|
| 1 | Subnet mask |
| 3 | Gateway |
| 6 | DNS |
| 12 | Hostname |
| 15 | Domain name |
| 43 | Cisco WLAN controller |

| | |
|---|---|
| 51 | Lease time |
| 58 | Renew time (half of lease time) |
| 66 | TFTP server |
| 82 | Relay agent information option |
| 150 | TFTP server for cisco phones |

**lease** *infinite* | *day hour min*

Default lease time is 24 hours (**lease 1 0 0**)

DHCP Server

```
interface vlan 10
 ip add 10.0.0.1 255.255.255.0

service dhcp
no ip dhcp conflict logging
ip dhcp excluded-address 10.0.0.1 10.0.0.10
ip dhcp excluded-address 10.0.0.240 10.0.0.254
ip dhcp pool DHCP
 network 10.0.0.0 /24
 default-router 10.0.0.1
 dns-server 10.0.0.10
 domain-name lab.local
 lease 1 0 0
```

Alternative

```
ip dhcp pool DHCP
 network 10.0.0.0 /24
 option 3 ip 10.0.0.1
 option 6 ip 10.0.0.10
 etc..
```

**clear ip dhcp binding** * | *ip-address*

```
show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
10.0.0.11           0063.6973.636f.2d63.    Jan 08 2017 09:13 AM    Automatic
                    6130.312e.3035.3734.
                    2e30.3030.382d.4661.
                    302f.30
```

```
show ip dhcp pool
Pool DHCP :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 254
 Leased addresses               : 1
 Excluded addresses             : 25
 Pending event                  : none
 1 subnet is currently in the pool :
 Current index        IP address range                    Leased/Excluded/Tota
 10.0.0.12            10.0.0.1         - 10.0.0.254        1     / 25    / 254
```

## DHCPv4 Client

- Cisco routers can function as a dhcp client and automatically install options such as DNS servers and default gateways
    - The default gateway will be configured as a static route to 0.0.0.0/0 with an administrative distance of 254
    - This gateway is installed even when **ip routing** is enabled
- This behavior can be overridden by disabling the DHCP options that are requested

```
interface gi0/0
 ip address dhcp
 no ip dhcp-client broadcast-flag
 no ip dhcp client request router
 no ip dhcp client request dns-nameserver
 no ip dhcp client request domain-name
```

**broadcast-flag** - Request unicast DHCPOFFER and DHCPACK from the server (default is broadcast)

**release dhcp** *interface-id*

```
show dhcp lease
Temp IP addr: 10.0.0.11  for peer on Interface: GigabitEthernet0/0
Temp  sub net mask: 255.255.255.0
   DHCP Lease server: 10.0.0.1, state: 5 Bound
   DHCP transaction id: 16CB
   Lease: 86400 secs,  Renewal: 43200 secs,  Rebind: 75600 secs
Temp default-gateway addr: 10.0.0.1
   Next timer fires after: 11:57:44
   Retry count: 0   Client-ID: cisco-ca01.0574.0008-Fa0/0
   Client-ID hex dump: 636973636F2D636130312E303537342E
                       303030382D4661302F30
   Hostname: CR1
```

# Relay (Helper)

## DHCPv4 Relay Agent

- Relay agents receive DHCP messages from clients and relay them to DHCP servers in different subnets
- The original DHCP message is regenerated and sent to the address specified with the **ip helper-address**
    - Configure the helper address on the interface (or VLAN) where DHCP messages arrive

- The DHCP relay agent sets the gateway address (giaddr) and adds the relay agent information option (option 82) to the forwarded packet
    - The giaddr address is used by the dhcp server to identify which subnet (VLAN) the request originated on
    - Using this information, the server knows which pool to assign an address from
- Option 82 adds additional information such as the remote-id, which is just a text string received on the server
    - Adds information of hosts such as mac-address and the switchport the request originated from, disabled by default
    - This information can be used to set QoS, ACL or security policies
    - Can cause issues with DHCP snooping

A DHCP relay agent may receive a message from another DHCP relay agent that already contains relay information (relayed twice)

- By default, the relay information from the previous agent is replaced
    - Customize with **ip dhcp relay information policy** command
    - If the information policy is changed, also disable the information check (**no ip dhcp relay information check**)

```
service dhcp
ip dhcp relay information option

interface vlan 10
 ip address 10.0.0.1 255.255.255.0
 ip helper-address 172.16.0.100
 ip helper-address 192.168.0.100
```

```
debug ip dhcp server packets
DHCPDISCOVER received from client 0063.6973.636f.2d63.6130.312e.3035.3734.2e3
Sending DHCPOFFER to client 0063.6973.636f.2d63.6130.312e.3035.3734.2e30.3030.
```

You can configure multiple ip helper-addresses per interface

- Each DHCPOFFER / DHCPREQUEST sent by clients will be relayed to each dhcp server
- If both dhcp servers sent offers they will both be forwarded to the client, which will choose the address
- The client will reply to both servers with a DHCPREQUEST
- The server that is not chosen will receive a request with DHCP option 50, which includes the address (from the other server) that the client has chosen

The **ip helper-address** command does not just forward DHCP messages

- Also forwards UDP broadcasts, including DHCP/BOOTP messages

Forwards:

- TFTP(UDP 69)
- DNS (UDP 53)
- Time protocol (UDP 37)
- NetBIOS name server (UDP 137)
- NetBIOS datagram server (UDP 138)
- Bootstrap Protocol / DHCP (UDP 67)
- Tacacs (UDP 49)

## DHCPv4 Information Option 82

- DHCP option 82 allows DHCP relay routers to inform the DHCP server where the original request came from
- This option is important when DHCP snooping is also being used in the network (see switching section)
    - Option 82 identifies hosts by both the mac-address and the switchport
    - Disabled by default on routers, enabled by default on switches (not all)
    - The reply from the server is forwarded back to the client after removing option 82
- Enable with **ip dhcp relay information option** globally or on interface
    - Interface configuration takes preference over global configuration

# Reservation

## DHCPv4 Reservation

- Reservations are set using the **client-identifier**, not the mac-address
- Cisco routers use the client-identifier to identify themselves
  - This is a combination of the hardware address, interface name and the name cisco
  - This client-identifier is then turned into a HEX string and presented to the server
  - Client-identifiers hex string always begins with 00, if not present add 00 manually
- An easy way to acquire the client-identifier is by **debug ip dhcp server packets**

Add 01 to the client mac-address to get the client-identifier for ethernet

- Format of the client-identifier is different than the mac-address
  - Mac-address format is xxxx.xxxx.xxxx
  - Client-identifier format is xxxx.xxxx.xxxx.xx where the mac-address is shifted to the right

```
debug ip dhcp server packets
DHCPDISCOVER received from client 0063.6973.636f.2d63.6130.312e.3035.3734.2e36
```

DHCP Server

```
ip dhcp pool CR1
 client-identifier 0063.6973.636f.2d63.6130.342e.3233.6130.2e30.3030.382d.4769.302f.30
 host 10.0.0.101 /24
 dns-server 10.0.0.10
 default-router 10.0.0.1
```

```
show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
10.0.0.101          0063.6973.636f.2d63.    Infinite                Manual
                    6130.312e.3035.3734.
                    2e30.3030.382d.4661.
                    302f.30
```

# IRDP

## ICMP Router Discovery Protocol (IRDP)

- Allows routers to advertise themselves as a gateway
- Similar in concept to IPv6 RAs
- IP routing has to be disabled on client

**Server**

```
int se0/0
 ip irdp
 ip irdp maxadvertinterval 30
 ip irdp minadvertinterval 10
 ip irdp holdtime 90
 ip irdp preference 200

show ip irdp se0/0
```

**Client**

```
no ip routing
ip gdp irdp
```

# FHRP

# GLBP

| **Protocol** | **HSRP** | **VRRP** | **GLBP** |
|---|---|---|---|
| Priority | Default 100<br>Range 0-255 | Default 100<br>Range 1-254 | Default 100<br>Range 1-255 |
| Preemption | Default disabled | Default enabled | AVG disabled<br>AVF enabled |
| Groups | 0-255<br>Group 0 is default | 1-255<br>No default group | 0-1023<br>No default group |
| Port | UDP 1985 (v1 / v2)<br>UDP 2029 (v6) | IP 112 | UDP 3222 |
| Multicast | 224.0.0.2 (v1)<br>224.0.0.102 (v2)<br>FF02::66 (v6) | 224.0.0.18 (v2)<br>FF02::12 (v3) | 224.0.0.102<br>FF02::66 (v6) |
| Mac-address | v1 = 0000.0c07.ac**XX**<br>v2 = 0000.0c9f.f0**XX** | 0000.5e00.01**XX** | 0007.b400.**XXYY** |

| | | | |
|---|---|---|---|
| | v6 = 0005.73a0.00**XX** | | |
| Timers | Hello 3<br>Hold 10 | Hello 1<br>Hold = Hello * 3 | Hello 3<br>Hold 10 |
| Max per group | 2 | 2 | 4 |

**XX** = Group number in HEX
**YY** = GLBP AVF number

## Gateway Load-Balancing Protocol (GLBP)

- There is one AVG (the master) per GLBP group and up to 3 other AVFs
  - The AVG is also an AVF so the maximum amount of AVFs (GLBP speakers) is 4
- GLBP does not allow you to track interfaces (unlike HSRP)
- The only form of tracking with GLBP is done through the weighted load-balancing method (see below)

GLBP States

| **State** | **State #** | **Definition** |
|---|---|---|
| Disabled | 0 | Interface is down and GLBP is not running |
| Initial | 1 | Interface comes online but GLBP is not yet running, primary state |
| Learn | 2 | Device is trying to learn the virtual ip-address (virtual-ip is not configured)<br>Device is waiting to hear from the neighbor device<br>This state will only be present if standby … options is configured, but the virtual ip-address is not |
| Listen | 3 | Device has learned the virtual ip-address<br>Listens for hellos from the neighbor device<br>Device is still waiting to hear from neighbor device in order to move to active or standby state |
| Speak | 4 | Device is sending periodic hellos and is participating in the active/standby election<br>Device cannot enter this state if it does not have a virtual ip-address |
| Standby | 5 | Neighboring device has won the election and is the AVG<br>Device is waiting to become the AVG if the neighbor fails<br>Device functions as AVF and forwards packets sent to the virtual mac-address |
| Active | 6 | Device is the active virtual gateway (AVG) and forwards packets sent to the virtual mac-address |

| | | |
|---|---|---|
| | | Device will first move to the standby state before moving on the active state |

## GLBP Priority / Preemption

```
glbp group_number priority 0-255
glbp group_number preempt delay minimum seconds
glbp group_number forwarder preempt delay minimum seconds
```

- Change the priority (100 by default)
  - Higher priority is better
  - Without preemption, the priority value only has effect when the GLBP group first establishes
- AVG preemption is disabled by default
  - Optionally set a delay value in seconds to wait before preemption (this can prevent flapping)
- AVF preemption is enabled by default with a 30 second delay

## GLBP Authentication

```
glbp group_number authentication text string
```

- Authenticate via a plain-text key set directly on the group

```
glbp group_number authentication md5 key-chain chain
glbp group_number authentication md5 key-string string
```

- Authenticate via a MD5 hash set directly on the group or via a key chain

```
interface vlan 10
 glbp 1 ip 10.0.0.101
 glbp 1 authentication md5 key-chain GLBP

key chain GLBP
 key 1
  key-string cisco
```

## GLBP Load-Balancing, Weighting & Tracking

| Method | Description |
|---|---|

| | |
|---|---|
| Host-dependent | Same AVF is used for the same host every time (based on host mac-address) |
| Round-robin (default) | Each AVF is used in turn<br>AVF1 first then AVF2, AVF3, AVF4 and back to AVF1 |
| Weighted | The AVF is chosen based on the set weighting value (all AVF are set to 100 by default) |

glbp *group_number* load-balancing **weighted | host-dependent | round-robin**

- You can set different load-balancing methods for individual groups

```
interface vlan 10
 glbp 10 ip 10.0.0.100
 glbp 20 ip 10.0.0.200

 glbp 10 load-balancing weighted
 glbp 20 load-balancing round-robin
```

```
show glbp

Vlan10 - Group 10
  Weighting 100 (default 100), thresholds: lower 1, upper 100
  Load balancing: weighted

Vlan10 - Group 20
  Weighting 100 (default 100), thresholds: lower 1, upper 100
  Load balancing: round-robin
```

Weighted load-balancing

- This is the only form of GLBP that offers tracking options
- AVF is chosen based on the set weighting value
  - Default value is 100 for all AVFs
  - AVFs can lose their forwarder status if they drop below a certain weight (based on tracking)
- Tracking objects (no interfaces/VLANs) can be used to decrement the weighting value
- The weight is basically just a percentage of the host requests that are forwarded to the AVF

glbp *group_number* weighting **maximum** *weight_value* **upper | lower** *weight_value*

| Weight | Description | Default value |
|---|---|---|
| Maximum | Normal (default) weighting value | 100 |
| Lower | If speaker goes below the lower weighting value, they lose their AVF status | 1 |
| Upper | If speaker goes above the upper weighting value, they regain their AVF status (after losing it) | Same as maximum |

Track three loopbacks in order to influence weighted value

```
track 1 interface lo1 line-protocol
track 2 interface lo2 line-protocol
track 3 interface lo3 line-protocol

interface vlan 10
 glbp 10 load-balancing weighted
 glbp 10 weighting 100 lower 51 upper 74
 glbp 10 weighting track 1 decrement 25
 glbp 10 weighting track 2 decrement 25
 glbp 10 weighting track 3 decrement 25
```

- Speaker will lose AVF status if only 1 tracking object is up
- Speaker will regain AVF status if 2/3 tracking objects are up

```
show track
Track 1
  Interface Loopback1 line-protocol
  Line protocol is Up
    3 changes, last change 00:01:56
  Tracked by:
    GLBP Vlan10 10
Track 2
  Interface Loopback2 line-protocol
  Line protocol is Up
    1 change, last change 00:01:49
  Tracked by:
    GLBP Vlan10 10
Track 3
  Interface Loopback3 line-protocol
  Line protocol is Up
    1 change, last change 00:01:49
  Tracked by:
    GLBP Vlan10 10
```

```
show glbp

Vlan10 - Group 10
  Weighting 100 (configured 100), thresholds: lower 51, upper 74
    Track object 1 state Up decrement 25
    Track object 2 state Up decrement 25
    Track object 3 state Up decrement 25
  Load balancing: weighted
```

## GLBP Redirection & Timers

glbp *group_number* **timers** *hello_interval hold_timer*
glbp *group_number* timers **msec** *hello_interval hold_timer*

- The **msec** keyword is required to send sub-second hellos (up to 1 hello every 50 milliseconds)

glbp *group_number* **timers redirect** *redirect_timer hold_timer*

- Redirect timer is 10 minutes (600 seconds) by default
- Timeout timer is 4 hours (14400 seconds ) by default

In case of an unreachable AVF the AVG redirects traffic:

- During redirecting time, the AVG points a new AVF for any new request with old virtual mac-address
- After the **redirect timer** expires, the AVG stops pointing a new AVF for any new request with old virtual mac-address
    - Hosts that using old mac-address can get responses and are able to use old mac address until the **timeout timer** expires
    - If the AVF doesn't return until the timeout timer expires, all GLBP peers flush the record of the old mac-address and old AVF

## GLBP Configuration

```
access-list 100 permit ip any host 224.0.0.2
access-list 100 permit ip any host 224.0.0.102
access-list 100 permit udp any any eq 3222
access-list 100 permit ip any any

track 1 interface lo0 line-protocol

interface vlan 10
```

```
ip address 10.0.0.1 255.255.255.0
ip access-group 100 in

glbp 12 ip 10.0.0.100
glbp 12 priority 200
glbp 12 preempt delay minimum 10
glbp 12 weighting 200
glbp 12 load-balancing weighted
glbp 12 weighting track 1 decrement 50
```

```
show glbp

Vlan10 - Group 12
  State is Active
    1 state change, last state change 00:01:26
  Virtual IP address is 10.0.0.100
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.664 secs
  Redirect time 600 sec, forwarder timeout 14400 sec
  Preemption enabled, min delay 10 sec
  Active is local
  Standby is 10.0.0.2, priority 100 (expires in 7.680 sec)
  Priority 200 (configured)
  Weighting 200 (configured 200), thresholds: lower 1, upper 200
    Track object 1 state Up decrement 50
  Load balancing: weighted
  Group members:
    ca02.298c.0008 (10.0.0.1) local
    ca03.0658.0008 (10.0.0.2)
  There are 2 forwarders (1 active)
  Forwarder 1
    State is Active
      1 state change, last state change 00:01:19
    MAC address is 0007.b400.0c01 (default)
    Owner ID is ca02.298c.0008
    Redirection enabled
    Preemption enabled, min delay 30 sec
    Active is local, weighting 200
  Forwarder 2
    State is Listen
    MAC address is 0007.b400.0c02 (learnt)
    Owner ID is ca03.0658.0008
    Redirection enabled, 597.696 sec remaining (maximum 600 sec)
    Time to live: 14397.696 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
    Active is 10.0.0.2 (primary), weighting 100 (expires in 8.064 sec)
```

- Notice that the virtual mac-address is 0007.bf00.0c01 for AVF1 and 0c02 for AVF2
  - 0c is group 12
  - 01 is forwarder number 1

- 02 is forwarder number 2

```
show glbp brief

Interface   Grp  Fwd Pri State    Address         Active router   Standby rou
Vl10        12   -   200 Active   10.0.0.100      local           10.0.0.2
Vl10        12   1   -   Active   0007.b400.0c01  local           -
Vl10        12   2   -   Listen   0007.b400.0c02  10.0.0.2        -
```

- This is from the AVGs perspective, which lists itself as the active router for AVF number 1
- The neighbor (10.0.0.2) is AVF number 2

- The first line is the state of the AVG, which lists the active router as local (meaning that this speaker is the AVG)

# HSRP

| **Protocol** | **HSRP** | **VRRP** | **GLBP** |
|---|---|---|---|
| Priority | Default 100<br>Range 0-255 | Default 100<br>Range 1-254 | Default 100<br>Range 1-255 |
| Preemption | Default disabled | Default enabled | AVG disabled<br>AVF enabled |
| Groups | 0-255<br>Group 0 is default | 1-255<br>No default group | 0-1023<br>No default group |
| Port | UDP 1985 (v1 / v2)<br>UDP 2029 (v6) | IP 112 | UDP 3222 |
| Multicast | 224.0.0.2 (v1)<br>224.0.0.102 (v2)<br>FF02::66 (v6) | 224.0.0.18 (v2)<br>FF02::12 (v3) | 224.0.0.102<br>FF02::66 (v6) |
| Mac-address | v1 = 0000.0c07.ac**XX**<br>v2 = 0000.0c9f.f0**XX**<br>v6 = 0005.73a0.00**XX** | 0000.5e00.01**XX** | 0007.b400.**XXYY** |
| Timers | Hello 3<br>Hold 10 | Hello 1<br>Hold = Hello * 3 | Hello 3<br>Hold 10 |
| Max per group | 2 | 2 | 4 |

**XX** = Group number in HEX
**YY** = GLBP AVF number

## Hot Standby Router Protocol (HSRP)

- The virtual ip-address shared between two speakers is advertised via gratuitous ARP
  - The source-mac address of the ARP packet is the HSRP mac-address
  - The destination mac-addresses are the broadcast (ffff.ffff.ffff) and the Cisco proprietary address 0100.0ccd.cdcd
- Basically, the two speakers want to create a duplicate ip-address between themselves

HSRP States

| **State** | **State #** | **Definition** |
|---|---|---|
| Disabled | 0 | Interface is down and HSRP is not running |
| Initial | 1 | Interface comes online but HSRP is not yet running, primary state |
| Learn | 2 | Device is trying to learn the virtual ip-address (virtual-ip is not configured)<br>Device is waiting to hear from the neighbor device<br>This state will only be present if standby … options is configured, but the virtual ip-address is not |
| Listen | 3 | Device has learned the virtual ip-address<br>Listens for hellos from the neighbor device<br>Device is still waiting to hear from neighbor device in order to move to active or standby state |
| Speak | 4 | Device is sending periodic hellos and is participating in the active/standby election<br>Device cannot enter this state if it does not have a virtual ip-address |
| Standby | 5 | Neighboring device has won the election and is the active router<br>Device is waiting to become the active router if the neighbor fails<br>Device does not forward packets sent to the virtual mac-address |
| Active | 6 | Device is the active router and forwards packets sent to the virtual mac-address<br>Device will first move to the standby state before moving on the active state |

There are two sets of HSRP interface-level commands:

- **standby** … commands
- **standby** *group-number* … commands

The first set of commands mostly apply to group number 0 and can set 'global' HSRP settings
The second set of commands apply only to a specific group (0-255)

Global Interface HSRP Settings

- Apply to all groups

```
interface
 standby version 1-2
```

- Sets the HSRP version to 1 or 2 for all groups on the interface
  - Different interfaces can use different versions of HSRP
  - The default version is 1
  - IPv6 HSRP requires the configuration of **standby version 2**

```
interface
 standby use-bia
```

- Use the actual interface mac-address instead of the virtual HSRP mac-address

Group 0 Interface HSRP Settings

- Applies only to group 0

```
interface vlan 10
 ip address 10.0.0.1 255.255.255.0
 standby ip 10.0.0.100
 standby preempt
 standby priority 200
```

```
show run interface vlan 10

interface Vlan10
 ip address 10.0.0.1 255.255.255.0
 standby 0 ip 10.0.0.100
 standby 0 priority 200
 standby 0 preempt
```

```
show standby

Vlan10 - Group 0
State is Active
 2 state changes, last state change 00:02:10
Virtual IP address is 10.0.0.100
Active virtual MAC address is 0000.0c07.ac00 (MAC In Use)
`Local virtual MAC address is 0000.0c07.ac00 (v1 default)
Hello time 3 sec, hold time 10 sec
`Next hello sent in 1.712 secs
Preemption enabled
Active router is local
Standby router is unknown
Priority 200 (configured 200)
Group name is "hsrp-Vl10-0" (default)
```

## HSRP Timers

standby timers **msec** *hello_interval hold_timer*

standby *group_number* **timers** *hello_interval hold_timer*
standby *group_number* timers **msec** *hello_interval hold_timer*

- The first command only applies to group 0
- The **msec** keyword is required to send sub-second hellos (up to 1 hello every 15 milliseconds)

## HSRP Name

standby *group_number* **name** *hsrp_group_name*

- This is more than just a description and is used by **standby follow** and stateful NAT (see NAT section)

standby *group_number* **follow** *hsrp_group_name*

- The **standby** *group_number* **follow** command configures an HSRP group to become a client of another HSRP group
  - The master group needs to be defined with a name (**standby** *group_number* **name**) which the client links to
  - The main purpose of this is to limit HELLO messages when many groups exists on the link
  - Only the master group will send HELLOs and the clients alter their Active/Standby state based on the master

## HSRP MAC-Address

standby *group_number* **mac-address** *hsrp_mac_address*

- Change the virtual mac-address used by HSRP
  - Hardcoded addresses will show up as *cfgd* in **show standby**
  - Hardcoded addresses will show up as *v1/v2 default* in **show standby**

```
interface vlan10
 standby version 1
 standby ip 10.0.0.100
 standby 1 ip 10.0.0.101

 standby mac-address 0000.0000.0001
```

```
show standby
Vlan10 - Group 0
  Active virtual MAC address is 0000.0000.0001 (MAC In Use)
    Local virtual MAC address is 0000.0000.0001 (cfgd)

Vlan10 - Group 1
  Active virtual MAC address is 0000.0c07.ac01 (MAC In Use)
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
```

## HSRP Priority / Preemption & Tracking

standby *group_number* **priority** *0-255*
standby *group_number* **preempt delay minimum** *seconds*

- Change the priority (100 by default)
  - Higher priority is better
  - Without preemption, the priority value only has effect when the HSRP group first establishes
- Preemption is disabled by default
  - Optionally set a delay value in seconds to wait before preemption (this can prevent flapping)

standby *group_number* **track** *track_number* **<cr>**
standby *group_number* **track** *track_number* **shutdown**
standby *group_number* **track** *track_number* **decrement** *priority_value*

standby *group_number* **track** *interface/vlan* **<cr>**
standby *group_number* **track** *interface/vlan* **decrement** *priority_value*

Track an object or interface in order to decrement the priority or shutdown the group

- When you just press **<cr>** the actual configuration will be to decrement a priority of 10

Track the up/down state of an object to decrement the priority by 10

```
track 1 interface lo0 line-protocol

interface vlan 10
 standby 1 ip 10.0.0.101
 standby 1 priority 200
 standby 1 preempt

 standby 1 track 1
```

Track the up/down state of an object to decrement the priority by 50

```
interface vlan 10
 standby 1 ip 10.0.0.101
 standby 1 priority 200
 standby 1 preempt

 standby 1 track lo0 decrement 50
```

```
show track

Track 1  (via HSRP)
  Interface Loopback0 line-protocol
  Line protocol is Up
    2 changes, last change 00:00:02
  Tracked by:
    HSRP Vlan10 1
```

## HSRP Authentication

standby *group_number* authentication **text** *string*

- Authenticate via a plain-text key set directly on the group

standby *group_number* authentication **md5 key-chain** *chain*

```
standby group_number authentication md5 key-string string
```

- Authenticate via a MD5 hash set directly on the group or via a key chain

```
interface vlan 10
 standby 1 ip 10.0.0.101
 standby 1 authentication md5 key-chain HSRP

key chain HSRP
 key 1
  key-string cisco
```

## HSRP Configuration

Version 1 with master/followers and tracking

```
access-list 100 permit ip any host 224.0.0.2
access-list 100 permit ip any host 224.0.0.102
access-list 100 permit udp any any eq 1985
access-list 100 permit udp any any eq 2029
access-list 100 permit ip any any

track 1 interface lo0 line-protocol

interface vlan 10
 ip add 10.0.0.1 255.255.255.0
 ip access-group 100 in

 standby version 1
 standby ip 10.0.0.100
 standby priority 200
 standby preempt
 standby authentication md5 key-string cisco
 standby track 1
 standby name MASTER

 standby 1 ip 10.0.0.101
 standby 1 follow MASTER
 standby 2 ip 10.0.0.102
 standby 2 follow MASTER
 standby 3 ip 10.0.0.103
 standby 3 follow MASTER
```

```
show standby brief

Interface   Grp  Pri P State   Active          Standby         Virtual IP
Vl10        0    200 P Active  local           unknown         10.0.0.100
```

```
show standby

Vlan10 - Group 0
  State is Active
    2 state changes, last state change 00:00:02
  Virtual IP address is 10.0.0.100
  Active virtual MAC address is 0000.0c07.ac00 (MAC In Use)
    Local virtual MAC address is 0000.0c07.ac00 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.272 secs
  Authentication MD5, key-string
  Preemption enabled
  Active router is local
  Standby router is unknown
  Priority 200 (configured 200)
    Track object 1 state Up decrement 10
  Group name is "MASTER" (cfgd)
  Followed by groups:
    Vl10 Grp 1 Active 10.0.0.101 0000.0c07.ac01 refresh 10 secs (expires in 7.
    Vl10 Grp 2 Active 10.0.0.102 0000.0c07.ac02 refresh 10 secs (expires in 6.
    Vl10 Grp 3 Active 10.0.0.103 0000.0c07.ac03 refresh 10 secs (expires in 7.
```

IPv6 and IPv4 HSRP with standby version 2

```
interface vlan 10
 ip add 10.0.0.1 255.255.255.0
 ipv6 address fe80::1 link-local

 standby version 2
 standby 10 ip 10.0.0.100
 standby 20 ipv6 fe80::200

 standby 10 timers 2 6
 standby 20 timers 5 15
```

```
show standby

Vlan10 - Group 10 (version 2)
  State is Active
    2 state changes, last state change 00:00:41
  Virtual IP address is 10.0.0.100
  Active virtual MAC address is 0000.0c9f.f00a (MAC In Use)
    Local virtual MAC address is 0000.0c9f.f00a (v2 default)
  Hello time 2 sec, hold time 6 sec
    Next hello sent in 0.608 secs
  Preemption disabled
  Active router is local
  Standby router is unknown
  Priority 100 (default 100)
  Group name is "hsrp-Vl10-10" (default)

Vlan10 - Group 20 (version 2)
  State is Active
    2 state changes, last state change 00:00:06
  Link-Local Virtual IPv6 address is FE80::200 (conf)
  Active virtual MAC address is 0005.73a0.0014 (MAC In Use)
    Local virtual MAC address is 0005.73a0.0014 (v2 IPv6 default)
  Hello time 5 sec, hold time 15 sec
    Next hello sent in 3.072 secs
  Preemption disabled
  Active router is local
  Standby router is unknown
  Priority 100 (default 100)
  Group name is "hsrp-Vl10-20" (default)
```

# VRRP

| **Protocol** | **HSRP** | **VRRP** | **GLBP** |
|---|---|---|---|
| Priority | Default 100<br>Range 0-255 | Default 100<br>Range 1-254 | Default 100<br>Range 1-255 |
| Preemption | Default disabled | Default enabled | AVG disabled<br>AVF enabled |
| Groups | 0-255<br>Group 0 is default | 1-255<br>No default group | 0-1023<br>No default group |
| Port | UDP 1985 (v1 / v2)<br>UDP 2029 (v6) | IP 112 | UDP 3222 |
| Multicast | 224.0.0.2 (v1)<br>224.0.0.102 (v2)<br>FF02::66 (v6) | 224.0.0.18 (v2)<br>FF02::12 (v3) | 224.0.0.102<br>FF02::66 (v6) |

| | | | |
|---|---|---|---|
| Mac-address | v1 = 0000.0c07.ac**XX**<br>v2 = 0000.0c9f.f0**XX**<br>v6 = 0005.73a0.00**XX** | 0000.5e00.01**XX** | 0007.b400.**XXYY** |
| Timers | Hello 3<br>Hold 10 | Hello 1<br>Hold = Hello * 3 | Hello 3<br>Hold 10 |
| Max per group | 2 | 2 | 4 |

**XX** = Group number in HEX
**YY** = GLBP AVF number

## Virtual Router Redundancy Protocol (VRRP)

- The virtual ip-address shared between two speakers is advertised via gratuitous ARP
  - The source-mac address of the ARP packet is the VRRP mac-address
  - The destination mac-addresses are the broadcast (ffff.ffff.ffff) and the Cisco proprietary address 0100.0ccd.cdcd
- Basically, the two speakers want to create a duplicate ip-address between themselves
- Configurations and workings of HSRP and VRRP are very similar

VRRP States

| **State** | **State #** | **Definition** |
|---|---|---|
| Create | 0 | VRRP instance is defined |
| Disabled | 1 | Interface is down and VRRP is not running |
| Initial | 2 | Interface comes online but VRRP is not yet running, primary state |
| Backup | 3 | Neighboring device has won the election and is the master router<br>Device is waiting to become the master router if the neighbor fails<br>Device does not forward packets sent to the virtual mac-address |
| Master | 4 | Device is the master router and forwards packets sent to the virtual mac-address<br>Device will first move to the backup state before moving on the master state |

vrrp *group_number* timers **advertise** *1-255 seconds*
vrrp *group_number* timers **advertise msec** *50-999 milliseconds*
vrrp *group_number* timers **learn**

**advertise** - Set the hello-interval for VRRP

- The hold-timer is always calculated to be (roughly) 3 times the hello and cannot be configured directly

**advertise msec** - Set the hello-interval for VRRP in milliseconds
**learn** - Learn the timers from the other speaker (VRRP neighbor)

## VRRP Priority / Preemption & Tracking

- Tracking can decrement the priority by a set value (default is 10)
- Can only track objects, not interfaces or VLANs directly
    - Cannot shutdown the group based on a tracking state

```
vrrp group_number priority 1-254
vrrp group_number preempt delay minimum seconds
```

- Change the priority (100 by default)
    - Higher priority is better
    - Without preemption, the priority value only has effect when the VRRP group first establishes
- Preemption is enabled by default without a delay value
    - Optionally set a delay value in seconds to wait before preemption (this can prevent flapping)

```
vrrp group_number track track_number <cr>
vrrp group_number track track_number decrement priority_value
```

Track an object in order to decrement the priority

- When you just press **<cr>** the actual configuration will be to decrement a priority of 10

Track the up/down state of an object to decrement the priority by 10

```
track 1 interface lo0 line-protocol

interface vlan 10
 vrrp 1 ip 10.0.0.101
 vrrp 1 priority 200
 vrrp 1 preempt delay minimum 10

 vrrp 1 track 1
```

```
show track

Track 1
  Interface Loopback0 line-protocol
  Line protocol is Up
    1 change, last change 00:00:33
  Tracked by:
    VRRP Vlan1 1
```

```
show vrrp

Vlan10 - Group 1
  State is Master
  Virtual IP address is 10.0.0.101
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000 sec
  Preemption enabled, delay min 10 secs
  Priority is 200
    Track object 1 state Up decrement 10
  Master Router is 10.0.0.1 (local), priority is 200
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.218 sec
```

## VRRP Authentication

- Supports plain-text and md5 authentication (both key chains and key-strings)

vrrp *group_number* authentication **text** *string*

- Authenticate via a plain-text key set directly on the group

vrrp *group_number* authentication **md5 key-chain** *chain*
vrrp *group_number* authentication **md5 key-string** *string*

- Authenticate via a MD5 hash set directly on the group or via a key chain

```
interface vlan 10
 vrrp 1 ip 10.0.0.101
 vrrp 1 authentication md5 key-chain VRRP

key chain VRRP
 key 1
  key-string cisco
```

## VRRP Configuration

```
access-list 100 permit ip any host 224.0.0.18
access-list 100 permit 112 any any
access-list 100 permit ip any any

track 1 interface lo0 line-protocol

interface vlan10
 ip add 10.0.0.1 255.255.255.0
 ip access-group 100 in

 vrrp 10 ip 10.0.0.100
 vrrp 10 authentication text cisco
 vrrp 10 timers advertise 2

 vrrp 20 ip 10.0.0.200
 vrrp 20 preempt delay minimum 10
 vrrp 20 priority 200

 vrrp 20 track 1 decrement 50
```

```
show vrrp brief
Interface          Grp Pri Time  Own Pre State   Master addr     Group addr
vl10               10  100 6609      Y  Master  10.0.0.1        10.0.0.100
vl10               20  200 3218      Y  Master  10.0.0.1        10.0.0.200
```

```
show vrrp

Vlan10 - Group 10
  State is Master
  Virtual IP address is 10.0.0.100
  Virtual MAC address is 0000.5e00.010a
  Advertisement interval is 2.000 sec
  Preemption enabled
  Priority is 100
  Authentication is enabled
  Master Router is 10.0.0.1 (local), priority is 100
  Master Advertisement interval is 2.000 sec
  Master Down interval is 6.609 sec

Vlan10 - Group 20
  State is Master
  Virtual IP address is 10.0.0.200
  Virtual MAC address is 0000.5e00.0114
  Advertisement interval is 1.000 sec
  Preemption enabled, delay min 10 secs
  Priority is 200
    Track object 1 state Up decrement 50
  Master Router is 10.0.0.1 (local), priority is 200
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.218 sec
```

## VRRPv3

- IPv6 VRRP requires version 3 to be globally enabled

```
ipv6 access-list VRRP
 permit ipv6 any host FE02::66
 permit ipv6 any any

fhrp version vrrp v3

interface vlan 10
 ipv6 address fe80::1 link-local
 ipv6 traffic-filter VRRP in

 vrrp 10 address-family ipv6
  address fe80::100 primary
```

# IP SLA

TCP Connect

- Control messages communicates the port that will be used from the sender to the receiver (enabled by default)
- Disable control packets when using a well-known TCP port (telnet for example).
    - Using a well-known port does not need a responder on the receiver side
    - Using an unknown port requires a SLA responder at the destination

**Sender / Source**

```
ip sla 1
 tcp-connect 10.0.12.2 23 control disable
 threshold 500
 timeout 1000
 frequency 5
ip sla schedule 1 life forever start-time now
```

```
show ip sla statistics 1
IPSLAs Latest Operation Statistics
IPSLA operation id: 1
      Latest RTT: 36 milliseconds
Latest operation start time: 15:32:24 UTC Wed Nov 30 2016
Latest operation return code: OK
Number of successes: 5
Number of failures: 0
Operation time to live: Forever

show ip sla configuration 1
IP SLAs Infrastructure Engine-III
Entry number: 1
Owner:
Tag:
Operation timeout (milliseconds): 1000
Type of operation to perform: tcp-connect
Target address/Source address: 10.0.12.2/0.0.0.0
Target port/Source port: 23/0
Type Of Service parameter: 0x0
Vrf Name:
Control Packets: disabled
Schedule:
   Operation frequency (seconds): 5  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 500
```

```
Distribution Statistics:
  Number of statistic hours kept: 2
  Number of statistic distribution buckets kept: 1
  Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:
  Number of history Lives kept: 0
  Number of history Buckets kept: 15
  History Filter Type: None
```

**show tcp brief (receiver side)**

| TCB | Local Address | Foreign Address | (state) |
|---|---|---|---|
| 67FFF5F8 | 10.0.12.2.23 | 10.0.12.1.14740 | CLOSED |

## IP SLA Responder

- Does not calculate processing time, allowing for more accurate measurements on the speed of the link
- Enable globally with the **ip sla responder** command.
  - General IP SLA responder uses port 1967 for control messages (not important to remember)
  - Control messages are sent by the sender to provide the receiver with the port to listen to
- The receiver can also be configured to listen on a specific port for UDP-echo or TCP-connect
  - Configure this specific port with **ip sla responder udp-echo <port> <ip address>**
  - However this specific port must match on both sides and control messages should be disabled at the sender
  - It is not possible to use control messages at sender with a specific port at the receiver
  - It is also not possible to disable control messages at the sender and just configure **ip sla responder** at the receiver

**Sender / Source**

```
ip sla 1
 tcp-connect 10.0.12.2 2323 control disable

ip sla schedule 1 life forever start-time now
```

**Receiver / Destination**

```
ip sla responder tcp-connect port 2323
```

**show ip sla responder**

```
General IP SLA Responder on Control port 1967
General IP SLA Responder is: Disabled

Permanent Port IP SLA Responder
Permanent Port IP SLA Responder is: Enabled

tcpConnect Responder:
```

```
IP Address          Port
0.0.0.0             2323
::                  2323
```

## UPP Echo

- UDP echo always requires a responder at the destination
- If control messages are disabled, the responder must be configured to listen on the specific port

**Sender / Source**

```
ip sla 1
 udp-echo 10.0.12.2 5353
 threshold 500
 timeout 1000
 frequency 5
ip sla schedule 1 life forever start-time now
```

**Receiver / Destination**

```
ip sla responder
```

**show ip sla summary**

```
IPSLAs Latest Operation Summary
Codes: * active, ^ inactive, ~ pending

ID        Type      Destination       Stats       Return      Last
                                      (ms)        Code        Run
---------------------------------------------------------------------------------
*1        udp-echo  10.0.12.2         RTT=20      OK          1 second ago
```

## UDP Jitter

- Same as UDP echo, also requires a responder configured at the destination
- Success/failures will only be updated when all packets are analyzed (10 packets are sent by default)
    - Per-direction jitter (source to destination and destination to source)
    - Per-direction packet loss
    - Per-direction delay (one-way delay)
    - Round-trip delay (average round-trip time)

**Sender / Source**

```
ip sla 1
 udp-jitter 10.0.12.2 23232 num-packets 1000
  threshold 500
  timeout 1000
  frequency 5
```

```
 tos 184
ip sla schedule 1 start-time now life forever
```

## IP SLA Authentication

- The authentication hash is MD5
- Enabled for all SLAs present on the device
- Only applied on SLAs where both sides need to participate (using a responder)

```
key chain SLA
 key 1
  key-string cisco
ip sla key-chain SLA

show ip sla authentication
```

# uRPF

## Unicast Reverse Path Forwarding (uRPF)

- Verifies reachability of source address in packets being forwarded.
- Requires CEF, and the source address is checked based on the entry in the Forward Information Base (FIB)
- Use optional ACL to log dropped packets by uRPF, or use it to allow specific subnets that fail the check
    - ACLs are only checked when the regular uRPF check fails

uRPF Modes:

- With strict mode (**rx** keyword) packets must be received on the interface that the router uses to forward the return packet
- With loose mode (**any** keyword) it is only required that the source address appears in the FIB
    - Use loose mode when asymmetric routing paths are present in the network
- A third mode exists called 'VRF mode' that is not covered in depth in the CCNP/CCIE exams.
    - This mode is also referred to as uRPFv3 and is similar to loose mode while checking against the FIB for each specific VRF

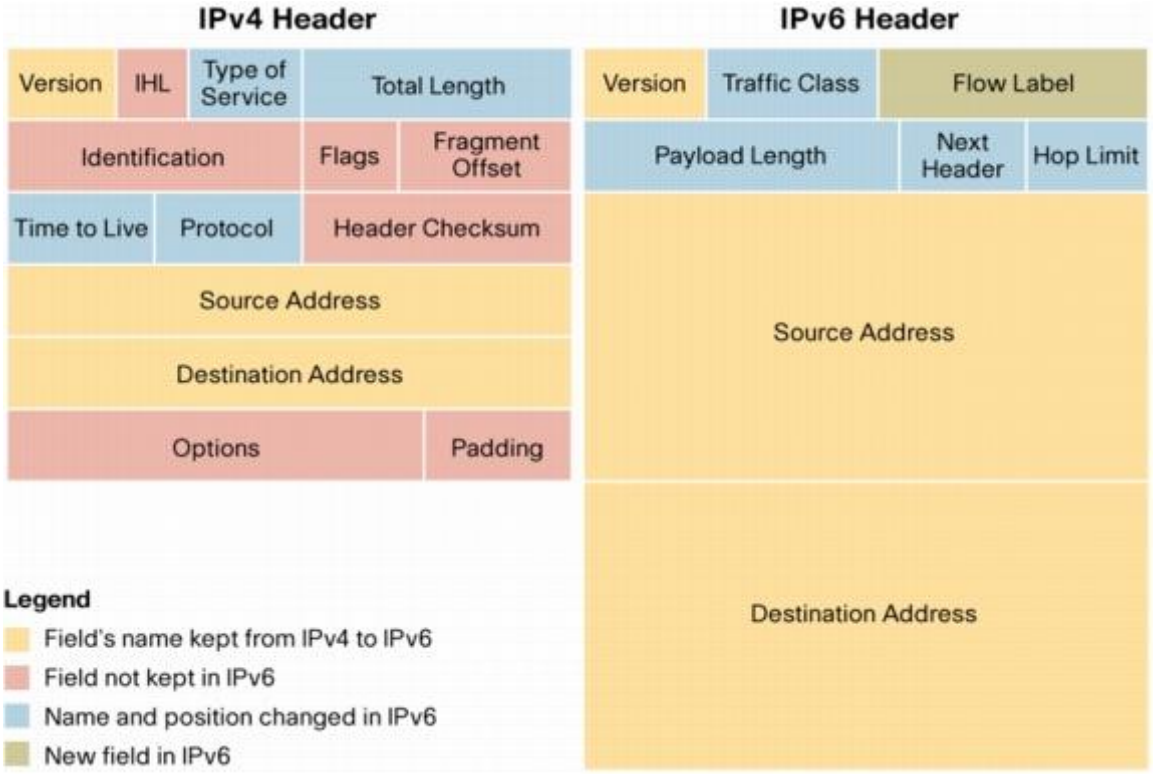
- The **allow-default** keyword also includes the default route in valid route list.
- The **allow-self-ping** keyword allows a router to ping itself on that particular interface.

```
access-list 100 permit ip 10.0.0.0 0.0.0.255 any

int fa0/0
 ip verify unicast source reachable-via any allow-default allow-self-ping
 ip verify unicast source reachable-via rx 100
```

**Verify**

```
show cef interface fa0/0
show ip interface fa0/0
```

# IPv6




# Address Ranges

IPv6 Address Ranges

| Prefix | Range | Definition | Most common | Information |
|---|---|---|---|---|
| 2000::/3 | 2000 - 3FFF | Global | 2001:: | Global unicast<br>Includes Registry-, Site-, and ISP (globally routable) prefixes |
| FE80::/10 | FE80 - FEBF | Link-Local | FE80:: | Local unicast / Link-local prefixes<br>Entire FE80::/10 range is available, but addresses are almost always FE80:: |
| FF00::/8 | FF00 - FFFF | Multicast | FF02:: | Usually FF02:: for all nodes / all routers multicast etc. |

| | | | | |
|---|---|---|---|---|
| | | | | FF05:: is often used for services such as DHCP and NTP |
| FC00::/7 **Deprecated** | FC00 - FDFF | Unique-Local | FD00:: | Unicast packets inside one organization<br>Entire FC00::/7 range is available, but addresses are almost always FD00:: |
| FEC0::/10 **Deprecated** | FEC0 - FEFF | Site-Local | FEC0:: | Similar in concept to RFC1918 IPv4 address space<br>Addresses are usually FEC0:: |

## IPv6 Global Unicast Assignments

| **Term** | **Mask** | **Assignment** |
|---|---|---|
| Registry prefix | ::/12 | IANA -> Regional Internet Registry (RIR) |
| ISP prefix | ::/32 | RIR -> ISP |
| Site prefix<br>Global routing prefix | ::/48 | ISP -> Customer (Enterprise) |
| Subnet prefix | ::/64 | Subnets inside Enterprise for individual links |

## IPv6 Common Multicast Addresses

| **Address** | **Purpose** | **Used for** | **IPv4 equivalent** |
|---|---|---|---|
| FF02::1 | All Nodes | Communication between all IPv6 enabled hosts<br>Used by NDP and most common IPv6 features | Broadcast 255.255.255.255 |
| FF02::2 | All Routers | Communication between routers only | - |
| FF02::5 | OSPF | OSPF HELLO and UPDATES | 224.0.0.5 |
| FF02::6 | OSPF DR | OSPF UPDATES in DR enabled networks | 224.0.0.6 |
| FF02::9 | RIPng | RIP UPDATES | 224.0.0.9 |
| FF02::A | EIGRP | EIGRP HELLO and UPDATES | 224.0.0.10 |
| FF02::1:FF | Solicited-Node | Duplicate Address Detection (DAD) | - |
| FF05::1:3 | DHCP | All DHCP servers (site scope) | - |
| FF05::101 | NTP | All NTP servers (site scope) | - |

## Anycast Address

- One to nearest form of communication
- An anycast address is shared by multiple systems, with the closest system being the receiver of the packet
- An address configured with :: at the end specifies an anycast interface

- The **anycast** keyword is optional and only required if there are multiple hosts on the same subnet
    - Meaning they are connected to the same interface on the router using the same address

```
int gi0/0
 description TO_SERVERS
 ipv6 address 2001:10:0:12::/64 anycast
```

## EUI-64 address

- An auto configured address using the MAC address of the interface
- A MAC address is split in the middle and a 16-bit hex value **FFFE** is inserted forming a 64-bit address
- Afterwards the **7th** bit is flipped in the OUI part of the MAC address

| **MAC address** | **MAC address + FFFE** | **Flip 7th bit in OUI** | **EUI-64 address** |
|---|---|---|---|
| A0:12:7A:CB:6B:40 | A0:12:7A:FF:FE:CB:6B:40 | 1010 0000 = A0<br>1010 0010 = A2 | A212:7AFF:FECB:6B40 |
| B6:14:9B:AA:12:FE | B6:14:9B:FF:FE:AA:12:FE | 1011 0110 = B6<br>1011 0100 = B4 | B414:9BFF:FEAA:12FE |

```
int gi0/0
 mac-address A012.7ACB.6B40
 ipv6 address 2001:10:0:12::/64 eui-64

Same as
int gi0/0
 ipv6 address 2001:10:0:12:A212:7AFF:FECB:6B40/64
```

# DHCPv6

## DHCPv6 multicast groups

| | |
|---|---|
| FF02::1:2 | Link-local DHCP using UDP 546, 547 |
| FF02::1:3 | Link-local multicast DNS using UDP 5355 |
| FF02::FB | Multicast DNS using UDP 5353 |
| FF05::1:3 | Site wide DHCP using UDP 546, 547 |

## DHCPv6 Messages

| **Message Type** | **Direction** | **Description** | **IPv4 Equivalent** |
|---|---|---|---|

| | | | |
|---|---|---|---|
| SOLICIT | Client -> Server | Sent by a client to locate servers | DHCPDiscover |
| ADVERTISE | Server -> Client | Sent in response to a SOLICIT | DHCPOffer |
| REQUEST | Client -> Server | Client requests an address or additional options | DHCPRequest |
| REPLY | Server -> Client | Sent in response to a SOLICIT, REQUEST, RENEW, REBIND<br>INFORMATION-REQUEST, CONFIRM, RELEASE, or DECLINE | DHCPAck |

- It is not possible to exclude addresses, or create reservations with DHCPv6
- DHCPv6 Lite is just stateless DHCPv6 (see below)

**Enable DHCPv6 on interface**

```
int gi0/0
 ipv6 dhcp server DHCP_POOL rapid-commit preference 255 allow-hint
```

- **rapid-commit** - Only SOLICIT and REPLY messages are used for address assignment
- preference - Higher preference will overrule other DHCPv6 servers on the link (default = 0)
- allow-hint - The server considers delegating client suggested prefixes (default = ignore)

```
show ipv6 dhcp interface GigabitEthernet0/0 is in server mode
  Using pool: DHCP_POOL
  Preference value: 255
  Hint from client: allowed
  Rapid-Commit: enabled
```

## DHCPv6 Stateful Mode

- DHCPv6 is used for address and other information (DNS / domain-name)
- Uses managed-config-flag (M-bit) and other-config-flag (O-bit)
    - Configuring **managed-config-flag** on interface is enough to use both M-bit and O-bit

**Server**

```
ipv6 dhcp pool STATEFUL
 address prefix 2001:10:0:12::/64
 dns-server 1::1
```

```
 domain-name lab.local

int gi0/0
 ipv6 dhcp server STATEFUL
 ipv6 nd managed-config-flag
```

**Clients**

```
int gi0/0
 ipv6 enable
 ipv6 address dhcp
```

```
show ipv6 dhcp pool
DHCPv6 pool: STATEFUL
  Address allocation prefix: 2001:10:0:12::/64 valid 172800 preferred 86400 (1
  DNS server: 1::1
  Domain name: lab.local
  Active clients: 1
```

```
show ipv6 dhcp binding
Client: FE80::A00:27FF:FE6E:1A6C
  DUID: 0004BDAA89EBF43519F8FA1A311A529A5533
  Username : unassigned
  IA NA: IA ID 0x276E1A6C, T1 1800, T2 2880
    Address: 2001:10:0:12:4545:F95F:77F1:F6A2
             preferred lifetime 172800, valid lifetime 86400
             expires at Nov 27 2016 06:59 PM (3313 seconds)
```

## DHCPv6 Stateless Mode

- Stateless Address Auto-Configuration (SLAAC) is used for address information
- DHCPv6 server is used for other information (DNS / domain-name)
- Uses only **other-config-flag** (O-bit)

**Server**

```
ipv6 dhcp pool STATELESS
 dns-server 1::1
 domain-name lab.local

int gi0/0
 ipv6 dhcp server STATELESS
 ipv6 nd other-config-flag
```

**Clients**

```
int gi0/0
 ipv6 address autoconfig default
```

- **default** - Also add the advertising (remote) router as the default-gateway even if **ipv6-unicast routing** is enabled

## IPv6 Default-Gateway

- The default gateway is a link-local address that is discovered through RAs
- Routers will only send out RAs if **ipv6 unicast-routing** is enabled
  - Routers will not receive a default-gateway if **ipv6 unicast-routing** is enabled
- Hosts normally pick the first router they discover through ND as the default-gateway
- This decision can be influenced with the router-preference command (default preference is medium)
  - Routers will preempt other gateways after decision has been made
- Optionally advertise the DNS server address through RAs (not supported on all platforms)

```
int gi0/0
 ipv6 nd router-preference high
 ipv6 nd ra dns server 1::1
```

Stop a router from becoming the default gateway with the **ipv6 nd ra lifetime 0** interface command

- This does not disable SLAAC or RAs, hosts will just not install a default route towards the router

Stop a router from sending RA messages and becoming the default gateway with the **ipv6 nd ra suppress all** interface command

- The **suppress** keyword indicates not to send periodic RAs
- The **all** keyword will instruct the router not to respond to RS

# Prefix-Delegation

## IPv6 Prefix-Delegation (PD)

```
ipv6 local pool LOCAL 2001:db8:abcd::/56 60
ipv6 dhcp pool PD_POOL
 prefix-delegation pool LOCAL lifetime infinite infinite

int fa0/0
 description TO_CLIENT
 ipv6 address 2001:db8:abcd::1/64
```

```
ipv6 dhcp server PD_POOL
```

**Client:**

```
int fa0/0
 description OUTSIDE
 ipv6 address autoconfig default
 ipv6 dhcp client pd PD_PREFIXES
int fa0/1
 description INSIDE
 ipv6 address PD_PREFIXES ::0:0:0:0:254/64
```

# Relay

## DHCPv6 Relay

- The interface on the DHCP router that the relay address points to has to be configured for a DHCP pool
- If relay destination is link-local, specify the outgoing interface

**Relay server**

```
int gi0/1
 description TO_CLIENTS
 ipv6 address 2001:10:0:12::254/64
 ipv6 nd managed-config-flag
 ipv6 dhcp relay destination fe80::1 gi0/0
```

**DHCPv6 server**

```
ipv6 dhcp pool STATEFUL
 address prefix 2001:10:0:12::/64
 dns-server 1::1
 domain-name lab.local

int gi0/0
 description TO_DHCP_RELAY
 ipv6 address fe80::1 link-local
 ipv6 dhcp server STATEFUL
```

# NDP

## Neighbor Discovery

- Uses ICMP and solicited-node multicast addresses to discover neighbors on the local link segment and verify reachability.
- A solicited-node multicast address starts with FF02:0:0:0:0:1:FF::/104.
- Is formed by taking the low-order 24 bits of an address and appending those bits to the solicited-node prefix.

- Afterwards, Duplicate Address Detection (DAD) sends a ping to the solicited node multicast address.
- If another node responds, then the router will not use the address.
- Is performed first on a new, link-local IPv6 address before the address is assigned to an interface.
- The new address remains in a tentative state [TEN] while DAD is performed.

| **Mac Address** | **Global Address** | **Link-Local Address** | **Solicited-Node Multicast Address** |
|---|---|---|---|
| a012.7acb.6b40 | 2001:10:0:12:a212:7aff:fecb:6b40/64 | fe80::a212:7a**ff:fe**cb:6b40 | ff02::1:ffcb:6b40/104 |
| b614.9baa.12fe | 2001:10:0:12:b414:9bff:feaa:12fe/64 | fe80::b414:9b**ff:fe**aa:12fe | ff02::1:ffaa:12fe/104 |

## Neighbor Solicitation and Router Advertisements

| Message Type | Description | Sent to |
|---|---|---|
| Neighbor Solicitation (NS) | Neighbor solicitations are used by nodes to determine the link layer address of a neighbor.<br>Or to verify that a neighbor is still reachable via a cached link layer address.<br>Also used in SLAAC (by hosts) to verify uniqueness of a local address before it is assigned. | FF02::1 |
| Neighbor Advertisement (NA) | Used by nodes to respond to a Neighbor Solicitation message. | FF02::1 |
| Router Solicitation (RS) | Requests neighbor address and advertises own,<br>Also used in SLAAC (by routers) to verify uniqueness of a local address before it is assigned. | FF02::1 |
| Router Advertisement (RA) | Periodically sent out between neighboring routers (200s default) on supporting interfaces<br>Used by routers to notify hosts of presence of a router and a default-gateway address<br>Can also be a reply to a RS message.<br>Will only be advertised if **ipv6 unicast-routing** is enabled | FF02::1 |

- RAs are disabled by default on P2P interfaces
    - No hosts (should) exist on these interfaces, so there's no point to send RAs

**Display the ND cache**

- This command will not display neighbors on P2P interfaces as there is no need to maintain a cache

```
show ipv6 neighbors
IPv6 Address                Age  Link-layer Addr   State   Interface
FE80::1                     6    0000.1111.1111    STALE   Gi0/0
```

## Important Neighbor States

| STATE | Description |
|---|---|
| INCMP | Incomplete<br>NS has been sent, but NA has not yet been received |
| REACH | Reachable |
| STALE | Reachable timer has expired |

**Define Static Neighbor**

```
ipv6 neighbor fe80::1 gi0/0 0000.1111.1111
```

```
show ipv6 neighbors
IPv6 Address                Age  Link-layer Addr   State   Interface
FE80::1                     -    0000.1111.1111    REACH   Gi0/0
```

Note that the state of a static neighbor is always REACH, unless the interface is down.

# Summarization

## IPv6 Address Summarization

| Addresses | Differences in Binary | Summary Range Start | Summary |
|---|---|---|---|
| 2001:db8:24:131a::<br>2001:db8:24:131b:: | 131a = 0001 0011 0001 1010<br>131b = 0001 0011 0001 1011 | 0001 0011 0001 101x = 131a<br>Subnets differ at 63th bit = /63 | 2001:10:0:12:24:131a::/63 |
| 2001:cfb:14::<br>2001:cfb:15::<br>2001:cfb:16::<br>2001:cfb:17:: | 0014 = 0000 0000 0001 0100<br>0015 = 0000 0000 0001 0101<br>0016 = 0000 0000 0001 0110 | 0000 0000 0001 01xx = 0014<br>Subnets differ at 46th bit = /46 | 2001:cfb:14::/46 |

| | | | |
|---|---|---|---|
| | 0017 = 0000 0000 0001 0111 | | |

# Tunneling

## Automatic Tunneling Methods

| | |
|---|---|
| Automatic 6to4 | Treats the underlying IPv4 network as an NBMA cloud.<br>Point-to-Multipoint. Uses 2002::/16 address space.<br>Encapsulates IPv4 address into IPv6 address (converted to HEX).<br>Does not support dynamic routing protocols. |
| Automatic IPv4 Compatible | Uses IPv4-compatible IPv6 addresses for the tunnel interfaces.<br>Point-to-Multipoint. Uses ::/96 address space (::IPv4-Address/96).<br>Deprecated. |
| ISATAP | Treats the underlying IPv4 network as an NBMA cloud.<br>Point-to-Multipoint. 0000:5EFE is the ISATAP address identifier.<br>Designed for tunneling within a site, not between sites.<br>Supports routing protocols using NBMA (neighbor statements). |

## Automatic 6to4

- The first 32-bits after the 2002::/16 address space as used for the converted IPv4 address.
- All addresses that need to be reachable over the tunnel need to be configured with the same 2002::/16 prefix.
- Alternatively, a route can be created for non 2002::/16 prefixes pointing to the tunnel address of the neighbor.

| IPv4 Address | Converted to HEX | 6to4 Address |
|---|---|---|
| 192.168.0.1 | c0.a8.00.01 | 2002:c0a8:0001::/64 |
| 172.16.30.254 | ac.10.1e.fe | 2002:ac10:1efe::/64 |
| 10.0.12.2 | 0a.00.0c.02 | 2002:0a00:0c02::/64 |

## 6RD

The 6RD feature is an extension of the 6to4 feature that uses encapsulation.

- Does not require 2002::/16 prefix or all 32bits of the IPv4 address.
- Embeds IPv6 into IPv4 using protocol type 41.

```
ipv6 unicast-routing
int se1/0
 ip address 10.0.123.1 255.255.255.0
int lo0
```

```
 ip address 192.168.0.1 255.255.255.255

int tun0
 ipv6 address 2002:c0a8:1::1/64
 tunnel source lo0
 tunnel mode ipv6ip 6to4

int lo1
 description PREFIXES_OVER_TUNNEL
 ipv6 address 2002:c0a8:1:1::1/64
 ipv6 address 2002:c0a8:1:2::1/64
 ipv6 address 2002:c0a8:1:3::1/64

ipv6 route 2002::/16 tunnel0
;ipv6 route 2::2/128 20002:c0a8:2::1
```

## ISATAP

- Automatically converts IPv4 addresses and inserts these in the IPv6 address.
- Preferably use eui-64 addressing. Free to choose network portion of the address.
- The last 32-bits after 0000:5EFE are used for the converted IPv4 address (in the host portion of the address)
- By default, tunnel interfaces disable periodic router advertisements (RA).
- RAs must be enabled on ISATAP tunnels to support client auto configuration. Enable with no ipv6 nd ra suppress.

| IPv4 Address | Converted to HEX | ISATAP Address with custom Network Range |
|---|---|---|
| 192.168.0.1 | c0.a8.00.01 | 2001:10:0:123:0000:5efe:c0a8:0001/64 |
| 172.16.30.254 | ac.10.1e.fe | 2001:10:0:123:0000:5efe:ac10:1efe/64 |
| 10.0.12.2 | 0a.00.0c.02 | 2001:10:0:123:0000:5efe:0a00:0c02/64 |

```
ipv6 unicast-routing
int fa0/0
 ip address 10.0.123.1 255.255.255.0
int lo0
 ip address 192.168.0.1 255.255.255.255

int tun0
 ipv6 address 2001:10:0:123::/64 eui-64
 tunnel source lo0
 tunnel mode ipv6ip isatap
 no ipv6 nd ra suppress
```

**Same as:**

```
int tun0
 ipv6 address fe80::5efe:c0a8:1 link-local
 ipv6 address 2001:10:0:123:0:5EFE:C0A8:1/64
 tunnel source lo0
 tunnel mode ipv6ip isatap
 no ipv6 nd ra suppress

router eigrp ISATAP
 address-family ipv4 autonomous-system 1
  network 192.168.0.1 0.0.0.0
  network 10.0.123.0 0.0.0.255
 address-family ipv6 autonomous-system 1
  neighbor fe80::5efe:c0a8:2 tun0
  neighbor fe80::5efe:c0a8:3 tun0

int lo1
 description PREFIXES_OVER_TUNNEL
 ipv6 address 1::1/128
 ipv6 address 11::11/128
```

## Static Tunneling Methods

- The only difference between GRE and IPv6IP configuration is the tunnel mode
- No need to configure a tunnel mode when using GRE

| | |
|---|---|
| Static IPv6IP | Carries only IPv6 packets over IPv4.<br>Point-to-Point. Uses protocol 41. |
| Static GRE | Carries IPv6, CLNS + other traffic over IPv4.<br>Point-to-Point. Uses protocol 47. (default method) |

```
ipv6 unicast-routing
int se1/0
 ip add 10.0.12.1 255.255.255.0

int tun0
 tunnel source se1/0
 tunnel destination 10.0.12.2
 ipv6 address fe80::1 link-local
 ipv6 address 2001:10:0:12::1/64
 tunnel mode ipv6ip
```

# MPLS

## MPLS Label Binding

- MPLS assigns a local label and receives a remote label for routes.
- The local label is distributed to neighbors and stored in their forwarding table.
- Traffic arriving with a label matching a local label can then immediately be forwarded to the remote label that is assigned for the same route, not requiring any form of lookup.

## Forwarding Equivalence Class (FEC)

- A label represents a FEC which is a group of IP packets which are forwarded in the same manner, over the same path.
- A FEC might correspond to a destination IP subnet or an IP precedence value.

## MPLS Infrastructure

- MPLS is globally enabled and requires CEF to operate.
- MPLS traffic will follow the same path as IP traffic by default.
- All IGPs (including connected and static) routes will have a label assigned by default.

## MPLS Label Operations

- PUSH. Label is installed on the packet, the label is pushed onto the stack.
- SWAP. The top-most label is swapped.
- PULL. The top-most label is removed from the stack.
- DELETE. The entire stack is deleted.

## MPLS Label Switching Routers (LSR) / Label Edge Router (LER)

- Traffic flows upstream to downstream in order to reach a network prefix.
- A MPLS downstream router is closest to the subnet that is being reached.
- Routers that reside in the core of the network are called LSRs.

Routers that reside connecting to CE devices are called Edge-LSRs (LER).

- Where traffic originating in the MPLS domain is called an Ingress Edge-LSR.
- The router that forwards the traffic to the CE is called an Egress Edge-LSR.

## Penultimate Hop Popping (PHP)

- The top-most label is removed by the LSR adjacent to the Edge-LSR.
- The label is popped one hop earlier than the Edge Egress LSR.

# L3VPN / IPVPN

## Route Distinguisher (RD) and Route Target (RT)

- The only purpose of the RD is to make routes unique in the mBGP.
- Doesn't have to match on neighboring routers that are part of the same VPNv4 neighborship.
- There is no relationship between RT and RD, they do not have to match on the same router or on neighboring router.
- RT helps sort routes in the appropriate routing table.
- The router imports the RT that the other router exports, this does not have to match the RD.

## Labels

- The top-most label (MPLS) is the transport label. This gets swapped between LSRs and popped at the egress PE.
- One transport label per VRF, not per route.
- The label between the top-most and the prefix is the mBGP label (VPN).
- One mBGP label per route in the VRF.

## L3VPN Configuration Steps

- Create VRFs and associate interfaces.
- MPLS and routing infrastructure is operational.
- Create VPNv4 infrastructure with mBGP peerings.
- Configure PE-CE routing.
- Configure mBGP -> PE-CE redistribution, this is not needed if PE-CE connection is using eBGP.

```
vrf definition 1
 rd 192.168.0.1:1
 address-family ipv4
  route-target both 1:1

router bgp 1
 no bgp default ipv4-unicast
 neighbor 192.168.0.2 remote-as 1
 neighbor 192.168.0.2 update-source Loopback0
 add vpnv4
  neighbor 192.168.0.2 activate
  neighbor 192.168.0.2 send-community both
```

## VPNv4 mBGP Peering Rules

- BGP needs a peering with a loopback address.
- If peered with physical address the PHP pops the label to soon because its a directly connected network.
- Fix non-loopback peering MPLS mBGP with route-map.
- A P router can also be configured as a RR. This will automatically disable the route-target filter.
- When using eBGP between VPNv4 peers the RT filter has to de explicitly disabled on the P router with the no bgp default route-target filter command.

```
route-map NEXT_HOP
 set ip next-hop 192.168.0.1
router bgp 1
 add vpnv4
  neighbor 10.0.12.2 route-map NEXT_HOP out
  neighbor 192.168.0.2 route-reflector-client
```

**VRF Route-Leaking between Sites:**

```
ip prefix-list Lo1 permit 11.11.11.11/32

route-map EXPORT_MAP permit 10
 match ip address prefix Lo1
 set extcommunity rt 11:11 additive

vrf definition 1
 rd 192.168.0.1:1
 address-family ipv4
  route-target both 1:1
  export map EXPORT_MAP

vrf definition 2
 rd 192.168.0.1:2
 address-family ipv4
  route-target both 2:2
  route-target import 11:11
```

# BGP

## BGP PE-CE

Either associate each CE with a different AS (65000+) or give each CE the same AS.
If same AS on CE:

- Configure CEs to allow their own AS inbound.
- Override CEs AS when forwarding BGP prefixes.
- Prevent loops between CE backdoors by setting the Site of Origin (SoO) on the PE

```
ip bgp-community new-format
router bgp 1
 no bgp default ipv4-unicast
 neighbor 192.168.0.3 remote-as 1
 neighbor 192.168.0.3 update-source Loopback0
 address-family vpnv4
  neighbor 192.168.0.3 activate
  neighbor 192.168.0.3 send-community both
 address-family ipv4 vrf 2
  neighbor 172.0.58.8 remote-as 65001
  neighbor 172.0.58.8 activate
  neighbor 172.0.58.8 send-community
  neighbor 172.0.58.8 as-override
  neighbor 172.0.58.8 soo 1:1

router bgp 65001
 address-family ipv4
  neighbor 172.0.58.5 remote-as 1
```

```
neighbor 172.0.58.5 allowas-in
```

### BGP Multipathing

- Same rules apply for normal multipathing.
- Routes from different PEs must have the same values in relation to cost, med, local-preference etc.
- When using full-mesh peering, the RD can be the same on all PEs.
- When using RR peering, the RD must be unique between the PEs that advertise the same routes.
- The route-import and export values can be the same between PEs.
- Specify the eibgp keyword otherwise multipathing is only applied for iBGP routes.

```
address-family ipv4 vrf 2
 maximum-path eibgp 32
```

# BGP GRE

### BGP GRE Tunnels (Old way)

- Alternative to using a MPLS / L3VPN configuration with VRFs:

```
int tun0
 ip address 13.0.0.1 255.255.255.0
 tunnel source lo0
 tunnel destination 192.168.0.3

route-map NEXT_HOP_TUNNEL
 set ip next-hop 13.0.0.1
router bgp 3
 neighbor 192.168.0.3 remote-as 3
 neighbor 192.168.0.3 update-source lo0
 address-family ipv4
 neighbor 192.168.0.3 route-map NEXT_HOP_TUNNEL out
```

# EIGRP / RIP

### EIGRP

- EIGRP routes that are redistributed into BGP receive a Cost Community ID of 128 by default.
- This will ensure that routes over the L3VPN are considered internal.
- This value is compared before all other BGP path selection attributes (including weight).
- The value/cost of the pre-bestpath community is the composite metric of the redistributed EIGRP route.
- Routes without this cost community are evaluated as if they had a cost value of 2147483647, which is half of the maximum possible value.

- Possible to modify the Cost Community ID. Lower values are better. Apply on EIGRP redistribution point or between VPNv4 neighbors.
- Ignore the cost community with the bgp bestpath cost-community ignore command.

```
router eigrp VPNV4
 address-family ipv4 unicast vrf 2 autonomous-system 1
 network 172.0.17.0 0.0.0.255
  topology base
   redistribute bgp 1 metric 100000 10 255 1 1500

router bgp 1
 add ipv4 vrf 2
  redistribute eigrp 1

ip prefix-list CE_Lo0 permit 192.168.0.6/32

route-map SET_EXT_COMM permit 10
 match ip address prefix CE_Lo0
 set extcommunity cost pre-bestpath 127 2662400
route-map SET_EXT_COMM permit 99

router bgp 1
address-family ipv4 vrf 2
 redistribute eigrp 1 route-map SET_EXT_COMM
```

## RIP

- Backdoor CE-CE using Offset-Lists
- An offset list of 7 in both directions will ensure that routes are not looped around the MPLS backbone.

```
router bgp 1
address-family ipv4 vrf 2
 redistribute rip route-map RIP

router rip
 address-family ipv4 vrf 2
  no auto
  version 2
  redistribute bgp 1 metric 1

router rip
 no auto
 version 2
 offset-list 0 in 7 FastEthernet0/0
 offset-list 0 out 7 FastEthernet0/0
```

# OSPF

## mBGP OSPF Super Backbone

The super backbone exists as the BGP cloud itself, as an area logically above Area 0.
If both CE routers connect to the PE are using OSPF area 0, the process ID that is configured on the CE-PE connection actually matters.

- If the process ID is different on the CEs the routes through the super backbone will be seen as external.
- If the process ID matches, the routes through the super backbone will be seen as inter-area routes.
- The OSPF domain ID is based on the process ID.
- If the CE's are using VRF-lite then it is required to disable the downward bit (D-bit) loop-prevention check when using the same domain-id.

```
router ospf 17 vrf 2
 network 172.0.17.0 0.0.0.255 area 0
 redistribute bgp 1 subnets
 domain-id 12.12.12.12
 capability vrf-lite

router bgp 1
 add ipv4 vrf 2
  redistribute ospf 26 vrf 2
```

## OSPF Backdoor CE-CE using Sham-Links

- The OSPF link through the MPLS cloud would be an inter-area link despite both CEs being in area 0.
- Internal routes are always preferred over inter-area and external routes, so all traffic will flow over the backdoor link.

Configuration steps:

- Create new loopback interfaces on PE routers.
- Associate loopback interfaces with VRF instance and assign unique IP.
- Advertise loopbacks in mBGP, but not in OSPF (Route-Map Filter).
- PE routers must be an ASBR, redistribute mBGP -> OSPF.
- Create sham-links on PE routers between new loopbacks.
- Modify cost on CE routers preferred internal interfaces.

Sham-links are Type-5 external LSAs. Networks sent over the sham-link are Type-1 LSA.

```
int lo1
 description SHAM LOOPBACK
 vrf forwarding 2
 ip add 1.1.1.1 255.255.255.255
```

```
router bgp 1
 address-family ipv4 vrf 2
  network 1.1.1.1 mask 255.255.255.255

ip prefix-list SHAM_LOOPBACK permit 1.1.1.1/32
ip prefix-list SHAM_LOOPBACK permit 2.2.2.2/32

route-map BLOCK_SHAM deny 10
 match ip address prefix-list SHAM_LOOPBACK
route-map BLOCK_SHAM permit 99

router ospf 17 vrf 2
 area 0 sham-link 1.1.1.1 2.2.2.2
 redistribute bgp 1 subnets route-map BLOCK_SHAM
```

# LDP / TDP

## Tag Distribution Protocol (TDP)

- Uses UDP 711 to create adjacency.
- Uses destination port TCP 711 to create session, random TCP source port.
- Does not support authentication.

TDP uses same commands as LDP. The only requirement is that it is either enabled globally or separately per interface.

- If not specified, LDP will be the default label protocol.
- TDP and LDP can coexist on the same router. However they must match on interfaces between neighbors.
- Interface configuration takes precedence over global configuration.
- TDP and LDP use a default hello timer of 5 seconds and a hold timer of 15 seconds.

## Label Distribution Protocol (LDP)

- Uses UDP 646 to create adjacency.
- Uses destination port TCP 646 to create session, random TCP source port.
- Supports authentication.

The LDP Router-ID is highest loopback by default followed by highest interface.

- The LDP RID is an actual IP address, unlike the BGP or OSPF RID.
- The LDP RID will be used as the transport address by default, meaning that it must be reachable.
- Between neighbors the LSR with the highest RID will initiate the TCP session.

Uses the all router multicast address 224.0.0.2 (UDP 646 or 711) to form the neighborship.

- Label Switched Paths (LSPs) are unidirectional.

```
ip cef
```

```
mpls ldp router-id loopback0 force
no mpls ip
mpls label range 100 199
mpls ip
int fa0/0
 mpls ip

show mpls ldp discovery detail
show mpls ldp neighbor detail
show tcp brief
```

## MPLS Transport Address

- The transport address can circumvent the RID for MPLS LDP neighbor advertisement.

```
int fa0/0
 mpls ldp discovery transport-address 10.0.12.1
```

## MPLS LDP Authentication

- If the session is already active, the password will have no effect until the session is cleared.
- Specifying the required keyword will require the local router to specify a password before neighborship can form.
- The LDP password applies to the RID, not the transport address. This RID must be reachable by the neighboring router.

```
mpls ldp neighbor 192.168.0.2 password cisco
mpls ldp password required
mpls ldp password rollover duration
```

A password rollover takes effect after the duration when passwords are configured without the use of a key chain.

- This feature is used when statically configured neighbor passwords (not with the option) need to be changed on the router.

```
ip access-list standard LDP_AUTH
 permit 192.168.0.0 0.0.0.255

key chain LDP
 key 1
  key-string cisco

mpls ldp password required
mpls ldp password option 1 for LDP_AUTH key-chain LDP
```

Globally configure password for all peers.

- Only used if neighbor password and password option are not configured

```
mpls ldp password fallback cisco
```

## Targeted Session

Can speed up label convergence time when the connection is restored after a failure.

- With a targeted session the session state between neighbors is kept after a failure,
- This is done by setting the holdtime to infinite.
- The targeted session can also be enabled globally for all peers or per prefix. Default duration is 24h.

```
mpls ldp neighbor 192.168.0.1 targeted ldp | tdp
mpls ldp session protection
```

## MPLS Filtering

**Only add labels for /32 prefixes (preferred):**

```
mpls ldp label
 allocate global host-routes

Other methods (not preferred):
ip prefix-list LOOPBACKS permit 192.168.0.1/32
ip prefix-list LOOPBACKS permit 192.168.0.2/32
ip prefix-list LOOPBACKS permit 192.168.0.3/32
ip prefix-list LOOPBACKS permit 192.168.0.4/32

mpls ldp label
 allocate global prefix-list LOOPBACKS
```

**Or:**

```
ip access-list standard LDP_ADV
 permit host 192.168.0.1
 permit host 192.168.0.2
 permit host 192.168.0.3
 permit host 192.168.0.4
no mpls ldp advertise-labels
mpls ldp advertise-labels for LDP_ADV

ip access-list standard LDP_REC
 permit host 192.168.0.1
 permit host 192.168.0.2
 permit host 192.168.0.3
 permit host 192.168.0.4
mpls ldp neighbor 192.168.0.2 labels accept LDP_REC
```

**Or, in the case of OSPF IGP:**

```
router ospf 1
 prefix-suppression

int fa0/0
 ip ospf prefix-suppression
```

## MPLS TTL

Hide the MPLS backbone by setting the TTL of traceroute traffic to 255.

- Forwarded applies to transit traffic.
- Local applies to locally generated traffic.
- Default is both.

```
no mpls ip propagate-ttl
```

**Handling of TTL expiring on packets:**

```
mpls ip ttl-expiration pop 1-6 (default is 0)
```

With 0 a packet with an expired TTL is forwarded by the global routing table.
With 1-6 a packet is forwarded by the underlying label, if more than 1 label is present.

**Allow the default route to be associated with a label:**

```
mpls ip default-route
```

# QoS

## MPLS Experimental Bits (EXP bits)

The DSCP value set on the IP packet is not changed by MPLS by default. Uses two modes:

- Pipe Mode. Uses egress queues based on EXP bits.
- Short Pipe Mode. Uses egress queues based on the 'original' ToS (DSCP) bits.

The DSCP value in the IP packet can also be replaced by the EXP bits, this is called 'Uniform Mode'.

## QoS Matching on PE using Groups

- On ingress interface, it is not possible to match on a IPP or DSCP value, because the MPLS header is still on the frame.
- On egress interface, it is not possible to match on the EXP bits to set IPP / DSCP bits, because the label is already popped.

The solution is to use QoS groups, which are local to the device itself.

- A packet is marked with a QoS group value only while it is being processed within the device.
- The QoS group value is not included in the packet's header when the packet is transmitted over the output interface.

```
class-map match-all EXP5
 match mpls experimental topmost 5
policy-map MPLS_INGRESS
 class EXP5
  set qos-group 5

int fa0/0
 description MPLS_CORE
 service-policy input MPLS_INGRESS

class-map match-all QOS_GROUP5
 match qos-group 5

policy-map MPLS_EGGRESS
 class QOS_GROUP5
  set ip dscp af41

int s1/0
 description VRF_CE
 service-policy output MPLS_EGGRESS
```

## Table-Maps

- Rewrite all ingress traffic using a Table Map.
- Table-Maps map a QoS group to a specific ToS value using DSCP values.

```
class-map match-all EXP5
 match mpls experimental topmost 5
policy-map MPLS_INGRESS
 class EXP5
  set qos-group 5

int fa0/0
 description MPLS_CORE
 service-policy input MPLS_INGRESS

table-map TABLE_MAP
 map from 1 to 8
 map from 2 to 16
 map from 3 to 24
 map from 4 to 32
 map from 5 to 40
 map from 6 to 48
 map from 7 to 56

policy-map MPLS_EGGRESS
```

```
 class class-default
  set dscp qos-group table TABLE_MAP

int s1/0
 description VRF_CE
 service-policy output MPLS_EGGRESS
```

## MPLS Implicit-null / Explicit-null

An implicit-null label is set to instruct upstream routers that they should perform PHP.

- The implicit-null label is set for directly connected prefixes on each LSR.

An explicit-null label is used in QoS combined with MPLS.

- When a packets gets encapsulated in MPLS, there is the option of copying the IP precedence to the MPLS header (EXP bits).
- If a POP is performed (implicit-null) at the penultimate LSR, the EXP bits in the MPLS header are removed as well.
- With explicit-null the MPLS header is left intact until it reaches the Egress LSR.
- Explicit Null is advertised in place of Implicit Null for directly connected prefixes.
- Configure with the mpls ldp explicit-null global command. Default is to enable explicit-null for all local prefixes.

**Limit explicit-null for route 10.10.10.0 only:**

```
ip access-list standard EXP_NULL
 permit host 10.10.10.0
mpls ldp explicit-null for EXP_NULL

Limit explicit-null to peer 192.168.0.2 only:
ip access-list standard EXP_NULL
 permit host 192.168.0.2
mpls ldp explicit-null to EXP_NULL
```

# Multicast

## Multicast Addressing

| | | |
|---|---|---|
| Link-Local | 224.0.0.0/24 | Used by network protocols on a local network segment. Non-Routable traffic. |
| Globally Scoped | 224.0.1.0 - 238.255.255.255 | Normal range. Can send between organizations and across the Internet. |
| SSM | 232.0.0.0/24 | Source-Specific Multicast (SSM) |

| | | |
|---|---|---|
| Private Multicast | 239.0.0.0/8 | Administratively scoped addresses. Equivalent of RFC1918 address space. |
| GLOP | 233.0.0.0/8 | Maps 16bit AS to multicast groups. |

### GLOP Addresses

- Convert AS to Hex, then take the separate parts of the 4-part hex and convert them back into decimal groups.
- Or convert straight to binary and separate the 16bit value into two groups of 8 and convert back into decimal.
- 0 and 255 are valid addresses in these ranges.

| AS | AS in binary | AS in hex | AS in decimal | GLOP address |
|---|---|---|---|---|
| 65053 | 11111110.00011101 | FE.1D | 254.29 | 233.254.29.0/24 |
| 64512 | 11111100.00000000 | FC.00 | 252.00 | 233.252.0.0/24 |

### Protocol Independent Multicast (PIM)

- Forms adjacencies with neighboring PIM routers.
- Default hello timer is 30 seconds, hold-time is 3x hello.
- PIMv2 hello uses IP protocol 103 and 224.0.0.13 (ALL-PIM-Routers address).

An interface can be configured in three different modes (RPF must succeed):

- Dense-Mode. Traffic is flooded on all enabled PIM interfaces.
- Sparse-Mode. Traffic is only forwarded only on interfaces with downstream clients. Uses RPs.
- Sparse-Dense-Mode. If RP is not known, operate in dense mode. If RP is known, operate in Sparse-Mode.

### Internet Group Messaging Protocol (IGMP)

- IGMP messages are sent in IP datagrams with IP protocol number 2, and a TTL of 1.
- IGMP packets pass only over a LAN and are not forwarded by routers, because of their TTL field values.
- IGMPv1. Clients join a multicast group.
- IGMPv2. Clients can also leave (all) groups. Backwards compatible with v1.
- IGMPv3. Clients can join and leave specific groups. Support for SSM.

# Anycast RP

### Anycast RP

- PIM Register and Join messages go to the closest RP in the topology.
- When PIM Register is received, MSDP Source Active (SA) is sent to MSDP peers which synchronize (S,G) information.
- The originator-id must point to a unique address on the router (do not use the same loopback used for the MSDP peering).

- The anycast loopback (not the peering loopback) is specified with the rp-candidate (BSR) or send-rp-announce (Auto-RP) command.

```
int lo0
 description MSDP_PEERING
 ip address 192.168.0.1 255.255.255.0255
 ip pim sparse-mode
int lo1
 description ANYCAST_RP
 ip address 12.12.12.12 255.255.255.255
 ip pim sparse-mode

ip msdp originator-id lo0
ip msdp peer 192.168.0.2 connect-source lo0
ip msdp mesh-group MSDP 192.168.0.2

ip pim bsr-candidate lo0
ip pim rp-candidate lo1

show ip msdp peer
show ip msdp sa-cache
debug ip msdp detail
debug ip msdp peer
```

### Exchange Multicast Information Without Anycast RP

- You do not necessarily need the same loopback IP when you want to exchange multicast information between multiple RPs.
- These RP's do not have the same IP address configured, instead it is just two separate RP's that will exchange multicast information.
- Can be used to link multicast areas together and have multiple RPs coexist with each other.

```
int lo0
 ip address 192.168.0.1 255.255.255.0255
 ip pim sparse-mode

ip msdp peer 192.168.0.2 connect-source lo0
ip pim bsr-candidate lo0
ip pim rp-candidate lo0
```

# Auto-RP

### Auto-RP Discovery and Announcements

- Auto-RP needs a RP to form multicast trees and allow traffic to flow.
- However the location of the RP has to be discovered through multicast as well.

- This creates a chicken and the egg situation. In order to find the RP, some kind of dense-mode solution is needed.
    - Statically assign the mapping agent and the RP for the 224.0.1.39-40 addresses. Kind of defeats the purpose of Auto-RP.
    - Configure interfaces with ip pim parse-dense mode. Uses dense mode for all groups without an RP, sparse for all others.
    - Configure the ip pim autorp listener. Allows usage of sparse-mode only interfaces and basically configures an ACL for the 224.0.1.39-40 addresses to be allowed to run in dense mode (preferred method).

## Mapping Agent

- Receive candidate messages (announcements) and decide which one will be the RP (Highest IP address wins).
- The mapping agents listen on 224.0.1.39 and propagate the decision to all other routers via 224.0.1.40.
- All routers join the 224.0.1.40 group by default, but only the mapping agents join 224.0.1.39.
- Configure with ip pim send-rp-discovery, the scope has be large enough to reach the DR for the PIM segments.

## Rendezvous Point (RP)

- Send multicast announcements (Dense Mode) to announce their RP candidacy to 224.0.1.39.
- Configure with ip pim send-rp-announce, the scope has be large enough to reach the mapping agent.
- If the mapping agent and the RP are the same router, a scope of 1 is enough.

```
ip pim autorp listener
ip pim send-rp-announce Loopback0 scope 255
ip pim send-rp-discovery Loopback0 scope 255

Static Auto-RP groups without listener (configure on all mrouters, including RP):
ip access-list standard AUTO_RP
 permit host 224.0.1.40
 permit host 224.0.1.39

ip pim rp-address 192.168.0.1 AUTO_RP
```

## Auto-RP Filtering

- Filter RP announcement messages on the mapping agent to only allow specific RPs, or bind RPs to specific groups.
- Configure on mapping agent only.

```
ip access-list standard RP1
 permit host 192.168.0.1
ip access-list standard RP2
 permit host 192.168.0.2
```

```
ip access-list standard GROUP_224_231
 permit 224.0.0.0 7.255.255.255

ip access-list standard GROUP_232_239
 permit 232.0.0.0 7.255.255.255

ip pim rp-announce-filter rp-list RP1 group-list GROUP_224_231
ip pim rp-announce-filter rp-list RP2 group-list GROUP_232_239

Deny all other RPs:
ip access-list standard OTHER_RP
 deny host 192.168.0.3
 deny host 192.168.0.4
 permit any

ip access-list standard GROUP_224_239
 permit 224.0.0.0 15.255.255.255

ip pim rp-announce-filter rp-list OTHER_RP group-list GROUP_224_239
```

## Auto-RP Cache Filtering

- Accept only (*, G) join messages destined for the specified Auto-RP cached address.
- Accept join and prune messages only for RPs in Auto-RP cache.
- Configure with the ip pim accept-rp auto-rp command on all mrouters.

# BSR

## Bootstrap Router (BSR)

- The rp-candidate is the actual RP. The bsr-candidate is the mapping agent
- Messages are flooded hop-by-hop by all multicast routers, this is because 224.0.0.13 is a link-local address.

Designed for Sparse-Mode, there is no need for Dense-Mode. Flooding is a control-plane feature and can be debugged.

- Auto-RP uses routable addresses that are outside the 224.0.0.0/24 range (224.0.1.39-40).
- BSR on the other hand uses link-local addresses, so its easier to control where traffic is flooded.
- Even though these messages are flooded, they are still subject to the RPF check.
- The edge of the BSR network can be specified with the ip pim bsr-border interface command.

```
ip access-list standard GROUP_224_231
 permit 224.0.0.0 7.255.255.255

ip pim rp-candidate Loopback 0 group-list GROUP_224_231
```

```
ip pim bsr-candidate Loopback 0

show ip pim bsr-router
debug ip pim bsr
```

# BIDIR-PIM

## Bidirectional PIM (BIDIR-PIM)

- No source-based (S,G) trees.
- RP builds a shared tree through which source routers forward traffic downstream toward the RP.
- The RP in BIDIR-PIM is always in the data plane, so placement is important.
- It's possible to limit which groups will be enabled for BIDIR, and use another or the same RP for regular ASM.

```
ip access-list standard BIDIR_RANGE
 permit 228.0.0.0 0.255.255.255
 permit 229.0.0.0 0.255.255.255

ip pim bidir-enable
ip pim bsr-candidate Loopback0
ip pim rp-candidate Loopback0 bidir BIDIR_RANGE
```

# PIM-DM

## PIM Dense Mode (PIM-DM)

- Forwards multicast traffic out of all interfaces, except the one received.
- Does not forward if no active downstream router and no hosts joined group.
- If both are true, router informs upstream router to stop sending via a prune message.
- If host joins the network after prune, then routers will use graft message to override prune.
- Only uses SPT.

State refresh messages can be used to 'refresh' the state before the 3min prune timer.

- This will stop all routers from pruning and un-pruning on the specified interval.
- Only has to be enabled on interface pointing to source, mrouter closest to source will relay state-refresh messages.
- Disable state-refresh with the ip pim state-refresh disable command.

```
int fa0/0
 ip pim state-refresh origination-interval 60
```

## PIM-DM Assert

- Prevents multiple senders from replicating the same multicast stream on to the wire.

- Used in dense-mode and enabled automatically.

In order to trigger a PIM Assert the (S,G) has to match exactly.
- IE both transferring routers need to connected to the same segment.
- Specifying a (loopback) source on the sender of the multicast traffic has no effect.

The winner is decided by:
- Lowest AD back to the source.
- In a tie, best metric value.
- In a tie, highest IP address.

# PIM-SM

## PIM Sparse Mode (PIM-SM)

- Assumes that no clients want to receive multicast packets until they specifically ask to receive them.
- Downstream routers request multicast traffic using PIM Join messages.
- Routers keep sending Joins, otherwise they are pruned.

## Rendezvous Point (RP)

- A common, agreed place in the network where clients can meet multicast sources.
- Not necessarily the center of the network.
- If there are many clients, it is better to make the router closest to the clients the RP.
- If there is only one source, it is better to make the router closes to the source the RP.
- All mrouters need to be configured with the location of the RP.

## RP Messages

- Multicast receivers (clients) inform the router that they want to receive multicast traffic, this is the (*,G) state.
- Multicast sources also inform the RP that they are sending multicast traffic, this is the (S,G) state (register message).
- This information is propagated through the network, by all routers that know the location of the RP.

- 3min state, after 3min the client will re-register with the RP. The RP informs routers to stop sending with register-stop message.
- As long as the source is transmitting this register-stop-register-stop state will continue
- This is similar to Dense-Mode, except that it is only between the RP and the source router (instead of all routers).

# RPF Failures / Fixes

## RPF Failures/Fixes

- Fix with static mroutes, multicast-BGP or tweaking unicast routing.

```
traceroute
mrinfo
show ip route multicast
show ip mfib
show ip mroute
show ip mroute count
show ip rpf
mtrace
debug ip pim
debug ip mfib pak
```

## Tunneling to fix RPF Failures

- Usage of loopback source/destination is preferred.
- Don't forget to enable PIM on the tunnel interface as well.

```
interface Tunnel 12
 ip address 12.0.0.1 255.255.255.0
 ip pim sparse-mode
 tunnel source Loopback 0
 tunnel destination 192.168.0.2

ip mroute 0.0.0.0 0.0.0.0 Tunnel 12
```

## Multicast-BGP to fix RPF Failures

- Static mroute is preferred over dynamic MBGP routes.
- Administrative distance of eBGP will make sure that MBGP routes are preferred over unicast routes (EIGRP or OSPF).
- Administrative distance of the IGP needs to be lowered on the router closest to the receiver in order for iBGP to be preferred.
- Advertise the source of the multicast traffic, and the location of the RP into MBGP on the router closest to the source.
- Works similar in concept to a static mroute, only the information is propagated by BGP.
- Advertise the network that needs to go over a different path instead of the unicast routing path.

Change the next_hop to the next BGP destination that the neighboring router must take.

- Remember that multicast will only try to go over PIM enabled interfaces.

Instruct R3 to choose R4 as the next-hop for mtraffic destined towards 172.16.0.0/24 (1.1.1.1 is the RP):

```
router bgp 234
 no bgp default ipv4-unicast
 neighbor 10.0.234.3 remote-as 234
 add ipv4 multicast
  network 172.16.0.0 m 255.255.255.0
  network 1.1.1.1 m 255.255.255.255
```

```
 neighbor 10.0.234.3 activate
 neighbor 10.0.234.3 route-map NEXT_HOP_MC out
 distance bgp 20 20 200

route-map NEXT_HOP_MC permit 10
  set ip next-hop 10.0.234.4
```

# SSM

## Source Specific Multicast (SSM)

- Does not require RP, BSR or Auto-RP.
- Receiver specifies the source address, RPF is still applied.
- Specify a custom range with the range statement (must fall within the 232.0.0.0/8 range).

**Configure router closest to receiver on same link:**

```
ip access-list standard SSM_RANGE
 permit host 232.0.0.1
 permit host 232.0.0.2

ip pim ssm range SSM_RANGE
int fa0/0
 description MULTICAST_SOURCE
 ip igmp version 3
 ip pim sparse-mode
```

**Configure receiver of multicast traffic:**

```
ip pim ssm default
int fa0/0
 description MULTICAST_RECEIVER
 ip pim-sparse mode
 ip igmp version 3

int lo0
 ip pim sparse-mode
 ip igmp join-group 232.0.0.1 source 192.168.0.1
 ip igmp join-group 232.0.0.2 source 192.168.0.1
 ip igmp join-group etc..
```

## SSM IGMP Filtering

**Configure router closest to receiver on same link to filter specific groups:**

```
ip access-list extended SSM_GROUPS
 permit igmp any host 232.0.0.1
```

```
int fa0/0
 ip igmp access-group SSM_GROUPS
```

# Static RP

## Static RP Configuration

- Statically configure each router with the location of the RP.
- This also has to be configured on the RP to point to itself.
- By default dynamically learned RP (BSR, Auto-RP) is preferred over static. Override this behavior with the override keyword.
- Between BSR and Auto-RP there is no preferred mapping order, the last mapping learned is preferred.

```
ip access-list standard GROUP_224_231
 permit 224.0.0.0 7.255.255.255

ip access-list standard GROUP_232_239
 permit 232.0.0.0 7.255.255.255

ip pim rp-address 192.168.0.1 GROUP_224_231 override
ip pim rp-address 192.168.0.2 GROUP_232_239 override
```

## RP Register Filtering

- Prevent unauthorized sources from registering with the RP (S,G). Configure on RP.
- If an unauthorized source sends a register message to the RP, the RP will immediately send back a register-stop message.

```
ip access-list standard GROUP_224_231
 permit 224.0.0.0 7.255.255.255

ip pim accept-register list GROUP_224_231
```

## RP Join Filtering

- Accept only (*, G) join messages destined for the specified RP address.
- Configure on mrouters, and optionally on RP.
- The group address must be in the range specified by the access list.
- If the RP points to itself, the RP will only accept registers from that particular multicast range.
- This is basically the same as the ip pim accept-register command.

```
ip access-list standard GROUP_224_231
 permit 224.0.0.0 7.255.255.255

ip access-list standard GROUP_232_239
```

```
 permit 232.0.0.0 7.255.255.255

ip pim accept-rp 192.168.0.1 GROUP_224_231
ip pim accept-rp 192.168.0.2 GROUP_232_239
```

### Dense-Mode Fallback

- Dense mode fallback allows the usage of dense mode if the RP becomes unreachable.
- Requires sparse-dense-mode configured on interfaces.

```
ip pim dm-fallback
int fa0/0
 ip pim sparse-dense-mode
```

# NAT

### Application Level Gateway (ALG)

- Some protocols embed IP address information in the Application Level payload.
- Regular NAT does not check the application level for protocols such as FTP, HTTP, DNS, SIP.
- ALG allows the use of dynamic ports by clients.
- ALG is on by default. Disable ALG by specifying the no-payload command.

### NAT Terms

- Static NAT (SNAT) - **One-to-One** translation of a single inside local address to a single inside global address
- Dynamic NAT (DNAT) - **Many-to-Many** translation of multiple inside local addresses to multiple inside global addresses (uses pools)
- Port Address Translation (PAT) - **Many-to-One** translation of multiple inside local addresses to a single inside global address (overload)
- Stateful NAT (SNAT) - Allows multiple routers to share NAT tables levering the HSRP FHRP protocol

### NAT Address Types

- Inside Global - A device inside of (our) network with a globally routable address
- Inside Local -A device inside of (our) network with a private address that needs NAT
- Outside Global -A device outside of (our) network with a globally routable address
- Outside Local - A device outside of (our) network with a private address that needs NAT

# NVI

### NAT Virtual Interface (NVI)

- No more concept of **nat inside** and **nat outside** interfaces

- The **add-route** keyword also adds the NAT_POOL route to the RIB, this can then be redistributed into BGP
- Using this method, the outside interface address does not necessarily have to match the NAT_POOL ip range

```
int gi0/0
 description PRIVATE
 ip address 10.0.0.1 255.255.255.0
 ip nat enable
int gi0/1
 description PUBLIC
 ip address 2.0.0.1 255.255.255.252
 ip nat enable

ip access-list standard NAT_ACL
 permit 10.0.0.0 0.0.0.255

ip nat pool NAT_POOL 1.0.0.1 1.0.0.2 prefix-length 30 add-route
ip nat source list NAT_ACL pool NAT_POOL

router bgp 64512
 neighbor 2.0.0.2 remote-as 2
 address-family ipv4
  network 1.0.0.0 mask 255.255.255.252
```

# Dynamic

## Dynamic NAT Pool

- Maps multiple inside local addresses to multiple inside global addresses (many-to-many)

```
ip access-list standard NAT_ACL
 permit 10.0.12.0 0.0.0.255

ip nat pool NAT_POOL 14.0.0.2 14.0.0.10 prefix-length 24
ip nat inside source list NAT_ACL pool NAT_POOL
```

## Dynamic NAT Pool using Route-Maps

- Use route-maps alongside dynamic NAT pools to provide more granular control.
- Can translate different traffic types to different outside addresses.

```
ip access-list extended NAT_ICMP
 permit icmp 10.0.12.0 0.0.0.255 any
ip access-list extended NAT_TCP
 permit tcp 10.0.12.0 0.0.0.255 any
```

```
ip access-list extended NAT_UDP
 permit udp 10.0.12.0 0.0.0.255 any

route-map NAT_ICMP_RM permit 10
 match ip address NAT_ICMP
route-map NAT_TCP_RM permit 10
 match ip address NAT_TCP
route-map NAT_UDP_RM permit 10
 match ip address NAT_UDP

ip nat pool NAT_POOL_ICMP 14.0.0.2 14.0.0.10 prefix-length 24
ip nat inside source route-map NAT_ICMP_RM pool NAT_POOL_ICMP

ip nat pool NAT_POOL_TCP 14.0.0.11 14.0.0.20 prefix-length 24
ip nat inside source route-map NAT_TCP_RM pool NAT_POOL_TCP

ip nat pool NAT_POOL_UDP 14.0.0.21 14.0.0.30 prefix-length 24
ip nat inside source route-map NAT_UDP_RM pool NAT_POOL_UDP
```

# PAT

## Port Address Translation

- Common NAT usage to map multiple Inside Local addresses to a single Inside Global address
- Only 65536 ports are available, extend the range by specifying more address in a pool
- The main identifier of a PAT configuration is the **overload** keyword

```
int fa0/0
 description PRIVATE_TO_R2
 ip address 10.0.12.1 255.255.255.0
 ip nat inside
int se1/0
 ip address 14.0.0.1 255.255.255.0
 description PUBLIC_TO_R4
 ip nat outside

ip access-list standard NAT_ACL
 permit 10.0.12.0 0.0.0.255

ip nat inside source list NAT_ACL int se1/0 overload
```

# Policy

## Policy NAT

- Uses tracking alongside route-maps in order to provide continuous NAT services on multiple outside interfaces.
- If main neighbor goes offline, remote the static route and switch over to the other neighbor.
- Poor man's NAT redundancy.
- Does not work with NAT pools.

```
int fa0/0
 description PRIVATE_TO_R2
 ip address 10.0.12.1 255.255.255.0
 ip nat inside
int se1/0
 description PUBLIC_TO_R3
 ip address 13.0.0.1 255.255.255.0
 ip nat outside
int se1/1
 description PUBLIC_TO_R4
 ip address 14.0.0.1 255.255.255.0
 ip nat outside

ip sla 1
 icmp-echo 13.0.0.3 source-interface se1/0
  frequency 5
track 1 ip sla 1

ip route 0.0.0.0 0.0.0.0 13.0.0.3 track 1
ip route 0.0.0.0 0.0.0.0 14.0.0.4 5

ip access-list standard NAT_ACL
 permit 10.0.12.0 0.0.0.255

route-map NAT_13 permit 10
 match ip address NAT_ACL
 match int se1/0
route-map NAT_14 permit 10
 match ip address NAT_ACL
 match int se1/1

ip nat inside source route-map NAT_13 interface se1/0 overload
ip nat inside source route-map NAT_14 interface se1/1 overload
```

# Rotary

## Rotary NAT (Round-Robin Load-Balancing)

- Can load-balance for servers located in the Private subnet range.
- This will only work for TCP traffic and the connections are forwarded in a round-robin (rotary) style to the inside network.
- If the outside connection is using ethernet interfaces, an ip alias needs to be created for the specific TCP traffic.
- Uses a destination list instead of a source list.

**Send Telnet traffic to R2 and R3 in a round-robin fashion:**

```
int fa0/0
 description PRIVATE_TO_R2_R3
 ip address 10.0.123.1 255.255.255.0
 ip nat inside
int fa0/1
 description PUBLIC_TO_R4
 ip address 14.0.0.1 255.255.255.0
 ip nat outside

ip alias 14.0.0.100 23
ip access-list standard ROTARY_NAT
 permit host 14.0.0.100

ip nat pool NAT_POOL 10.0.123.2 10.0.123.3 prefix-length 24 type rotary
ip nat inside destination list ROTARY_NAT pool NAT_POOL
```

# Static

## Static NAT

- Map a single Inside Local (IL) address to a single Inside Global (IG) address.
- The no-alias keyword will stop the creation of an alias for the global address space.
- The extendable keyword is added automatically in IOS, and is used when the same IL address is mapped to multiple IG addresses.

```
int fa0/0
 description PRIVATE_TO_R2
 ip address 10.0.12.1 255.255.255.0
 ip nat inside
int se1/0
 ip address 14.0.0.1 255.255.255.0
 description PUBLIC_TO_R4
 ip nat outside
```

```
ip nat inside source static 10.0.12.2 14.0.0.100 extendable
ip nat inside source static 10.0.12.2 14.0.0.200 extendable
```

## Static PAT

- Map specific ports to IG addresses (or the outside interface address).

```
ip nat inside source static tcp 10.0.12.2 23 int se1/0 2323
```

## Static NAT Network Range

- Map an entire IL network range to an IG range, preserving host portion addressing.
- This will create a static 1:1 range, meaning that 10.0.12.2 will be mapped to 14.0.0.2 for example.

```
ip nat inside source static network 10.0.12.0 14.0.0.0 /24
```

## Static NAT Pool

Configure a pool that contains IG addresses and map these to an access-list that contains the IL addresses.

```
ip access-list standard NAT_ACL
 permit 10.0.12.0 0.0.0.255

ip nat pool NAT_POOL 14.0.0.2 14.0.0.10 prefix-length 24
ip nat inside source list NAT_ACL pool NAT_POOL
```

## Reversible Static NAT Pool using Route-Maps

- Use route-maps alongside static NAT pools to provide more granular control.
- The reversible keyword enables outside-to-inside initiated sessions to use route_maps for destination-based NAT.

```
ip access-list extended NAT_ICMP
 permit icmp 10.0.12.0 0.0.0.255 any
ip access-list extended NAT_TCP
 permit tcp 10.0.12.0 0.0.0.255 any
ip access-list extended NAT_UDP
 permit udp 10.0.12.0 0.0.0.255 any

route-map NAT_ICMP_RM permit 10
 match ip address NAT_ICMP
route-map NAT_TCP_RM permit 10
 match ip address NAT_TCP
route-map NAT_UDP_RM permit 10
 match ip address NAT_UDP
```

```
ip nat pool NAT_POOL_ICMP 14.0.0.2 14.0.0.10 prefix-length 24
ip nat inside source route-map NAT_ICMP_RM pool NAT_POOL_ICMP reversible

ip nat pool NAT_POOL_TCP 14.0.0.11 14.0.0.20 prefix-length 24
ip nat inside source route-map NAT_TCP_RM pool NAT_POOL_TCP reversible

ip nat pool NAT_POOL_UDP 14.0.0.21 14.0.0.30 prefix-length 24
ip nat inside source route-map NAT_UDP_RM pool NAT_POOL_UDP reversible
```

# Stateful (SNAT)

## Stateful NAT (SNAT)

- The SNAT feature allows multiple routers to share NAT tables.
- When used alongside HSRP, the standby router can take over the NAT translations.
- The standby router can share state with the active router, keeping the NAT sessions alive.
- The mapping-id must be the same between peers. The redundancy string must match the standby name.
- The configuration of the standby router is identical, with the exception of the ip nat stateful id.

```
int fa0/0
 standby 1 name SNAT
 standby 1 ip 10.0.123.254
 ip nat inside
int se1/0
 ip nat outside

ip nat stateful id 1
  redundancy SNAT
   mapping-id 12

access-list standard NAT
 permit 10.0.123.0 0.0.0.255
ip nat pool NAT_POOL 12.0.0.100 12.0.0.100 prefix-length 24 add-route
ip nat inside source list NAT pool NAT_POOL mapping-id 12 overload

router bgp 12
 neighbor 14.0.0.4 remote-as 4
add ipv4
 network 12.0.0.0 mask 255.255.255.0

show ip snat peer 10.0.123.2
show ip snat distributed
```

# OSPF

## OSPF Networking

- Intra-area are preferred over inter-area routes.
- Intra-area and inter-area routes are preferred over external routes.
- E1 routes are preferred over E2, even though they might have higher cost.
- Type-3 LSAs received from the backbone area are never re-advertised into the backbone area.

- Internal routes (network statement) are advertised as Type-1 LSAs and are translated to Type-3 by ABRs.
- The next-hop of redistributed routes are advertised as Type-1 LSAs with the E-bit set by the ASBR and are translated to Type-4 LSAs by the next ABR.
    - The Type-4 LSA is updated by each ABR which changes the next-hop value.
- Redistributed routes are also advertised as Type-5 LSAs, this is the actual route, not the way to reach it.
- The Type-5 LSA next-hop is unchanged, Type-4 is used to reach the Type-5 next-hop.
- The Type-5 external LSA advertising router is not updated, only the Type-4 LSA indicating how to reach this prefix is updated.

## OSPF DR / BDR Election

- The DR/BDR election is not preemptive, the first router that boots will be selected.
- If all routers boot at the same time, the DR is chosen based on the highest priority, or the highest RID.
- The DR originates the network LSA on behalf of the network and forms adjacencies with all DROTHERS to synchronize the LSDB.
- BDR and DR perform the same function, except the BDR only sends out updates if the DR goes down.
- In NBMA networks the BDR is elected before the DR.

Primary DR Role

- All routing updates will be forwarded from the DROTHER routers to the DR/BDR using 224.0.0.6.
- The DR will update the LSAs and propagate the changes to the rest of the DROTHER routers using 224.0.0.5.

Secondary DR Role

- Informs all routers in the area of the shared segment using Type-2 LSA.
- This is only performed by the DR, not the BDR. The BDR only receives LSAs and is promoted if the DR fails.

## OSPF ABR / ASBR

- ABR routers must be connected to the backbone areas.
- ABRs are filtering and summarization point for inter-area routes.
- ABR summarizes Type-1 and Type-2 and Type-3 from other ABRs into Type-3 LSAs.
- Summarizes other received Type 3 information.
- ABR announce their ABR status using the Type-1 LSA other flag, the B-bit (Border-bit).

- ASBR set the E-bit to 1 (Edge-bit).

### OSPF Bits

- Bits that don't deal with NSSA areas or MPLS L3VPNs are usually set on Type-1 (router) LSAs

| Bit | Description | LSA Type |
|---|---|---|
| B | Border bit. | Type-1 |
| V | Virtual-link Endpoint. | Type-1 |
| E | Edge bit. Set by ASBRs upon injecting (redistributing) external routes into OSPF<br>Upon reception of a Type-1 LSA with the E-bit set, the ABR wil generate a Type-4 LSA | Type-1 |
| P | Propagate bit. The propagate bit will allow translation of Type-7 to Type-5<br>The **nssa-only** keyword will set the P-bit to 0, meaning that the route will stay in the NSSA area | Type-7 |
| F | Forward bit. Indicates that a forwarding address is included in the Type-7 LSA when translated to Type-5 | Type-7 |
| DN | Downward bit. Set by default only on routers running OSPF in a VR | Type-3 / Type-5 / Type-7 |

### OSPF Non-NSSA Forward Address

- The forward address (FA) is the highest address local enabled for OSPF. This is an actual ip-address not a RID.

If two routers are border routers, and only one is redistributing a route, then this router will be chosen as the next hop regardless of cost.
The above is true unless the following conditions are met:

- Exit interface pointing towards external destination must be enabled for OSPF and must not be passive.
- To be considered valid the external destination forward address must be known as OSPF route.
- The network attached to the external destination must be either broadcast or NBMA.

# Authentication

### OSPF IPv4 Authentication

- Authentication can be configured for an entire area, or per interface
  - Globally configuring authentication for area 0 will also enable authentication for virtual-links (are always in area 0)
- The key-strings themselves have to be configured per interface

- Interface authentication modes overrides area authentication
- When using MD5 the **key ID** and the **key-string** must match between neighbors
- Multiple keys can be configured on an interface, using this method key ID can be added/removed without any downtime (key rollover)
    - OSPF actually sends and accepts messages that use all the currently configured authentication keys on an interface
    - The youngest key ID appearing in the output of the **show ip ospf interface** is the one being used for authentication
    - The youngest key ID appearing in the output of the **show ip ospf interface** is the most recently configured, regardless of its key ID
        - Adding a new key ID will initiate a 'key rollover', during this time the neighborship will stay up until the dead-timer expires
        - Make sure to add the new key on both devices during the dead-timer, older keys can be deleted afterwards

```
R1#show ip ospf interface
Serial4/0 is up, line protocol is up
  Internet Address 10.0.13.1/24, Area 0, Attached via Network Statement
  Process ID 1, Router ID 10.0.13.1, Network Type POINT_TO_POINT, Cost: 64
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           64        no          no            Base
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:08
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 4 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.0.23.3
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 7
    Rollover in progress, 1 neighbor(s) using the old key(s):
      key id 1
      key id 2
      key id 3
      key id 4
      key id 5
      key id 6
```

OSPF Supports the following authentication types:

- Null (Type-0). Default.
- Plain-Text (Type-1). Simple authentication - has a **maximum key length** of 8 characters
- MD5 (Type-2). Message-Digest authentication - has a **maximum key length** of 16 characters

Technically OSPFv2 also supports SHA authentication, however this feature is not widely deployed in Cisco IOS. The same is true for IS-IS

**Simple Authentication (Type-1)**

```
router ospf 1
 area 0 authentication

int fa0/0
 ip ospf authentication-key cisco
 ip ospf authentication
```

**MD5 Authentication (Type 2)**

```
router ospf 1
 area 0 authentication message-digest

int fa0/0
 ip ospf message-digest-key 1 md5 cisco
 ip ospf authentication message-digest
```

## Virtual Link Authentication

- Virtual-Links are always considered to be area 0
- Virtual-Links use area 0 authentication

```
router ospf 1
 area 0 authentication message-digest
 area 1 virtual-link 192.168.0.2 authentication message-digest message-digest-key 1 md5 0 cisco
```

## OSPF IPv6 Authentication

- Requires the use of IPsec to enable authentication. Only supports full hexadecimal keys.
- To use the IPsec AH header, you use only the ipv6 ospf authentication command.
- When MD5 authentication is used, the key must be 32 hex digits long.
- When SHA-1 authentication is used, the key must be 40 hex digits long.

```
ipv6 router ospf 1
 area 0 authentication ipsec spi 256 md5 1234567890abcdef1234567890abcdef

int fa0/0
 ipv6 ospf authentication ipsec spi 256 md5 1234567890abcdef1234567890abcdef
```

## OSPF IPv6 Encryption

- Requires the use of IPsec to enable encryption. Only supports full hexadecimal keys.
- To use the IPsec ESP header, you use the ipv6 ospf encryption command.
- When ESP is set to a non-null value, both encryption and authentication are provided.
- It is not possible to configure encryption and authentication using different commands.

**ESP Null:**

```
ipv6 router ospf 1
 area 0 encryption ipsec spi 256 esp null md5 1234567890abcdef1234567890abcdef

int fa0/0
 ipv6 ospf encryption ipsec spi 256 esp null md5 1234567890abcdef1234567890abcdef
```

**ESP AES-CBC 128:**

```
ipv6 router ospf 1
 area 0 encryption ipsec spi 256 esp aes-cbc 128 1234567890abcdef1234567890abcdef md5
1234567890abcdef1234567890abcdef

int fa0/0
 ipv6 ospf encryption ipsec spi 256 esp aes-cbc 128 1234567890abcdef1234567890abcdef md5
1234567890abcdef1234567890abcdef
```

### OSPFv3 Authentication

- There is no built-in mechanism for authentication in OSPFv3, instead it relies on IPsec
- OSPFv3 (like OSPF IPv6) supports using IPsec for authentication
    - Both SHA and MD5 are supported hashing algorithms
- OSPFv3 does NOT support Type-1 plain text authentication

# Adjacencies / LSAs

### OSPFv2 Adjacency Requirements

- Matching hello/dead timers
- Matching area ID between neighbors on the same link
- Matching subnet mask and/or network prefix
- Matching area type (stub / transit / nssa)
- Matching authentication (type-0 / type-1 / type-2)
- Matching ospf network type (broadcast / p2p / p2mp / non-broadcast)
- Unique router-id (RID) between neighbors
- Matching MTU (1500)

### OSPFv3 Adjacency Requirements

- Matching Instance ID
- Above list except matching subnet, because of usage of link-local address

### OSPF Adjacency Process

- Adjacencies are formed using hellos
    - A router will send its own router-id (RID) in the hello message
    - A neighboring router will add the RID of received hellos to its own hello message
    - The original router will see its own RID in the neighbor's hello message, this proves that the hello was received by the neighbor
- OSPF routers form a neighborship with all routers on the link

- On broadcast and non-broadcast networks only FULL neighborships are created between the DR > BDR and the BDR/BDR > DROTHERS
- All other routers will create a 2-WAY neighborship (between different DROTHERS)

| **State** | **Description** | **OSPF Packet Types** |
|---|---|---|
| WAIT | Dormant state on broadcast and non-broadcast network types | |
| INIT | A hello message has been sent | Type-1 Hello |
| 2-WAY | Router has seen its own RID reflected in a neighbor's hello message<br>Possible stable state on DR/BDR network types | Type-1 Hello |
| EXSTART | Master/Slave relationship is formed and MTU is compared between neighbors<br>Decision is made on how to exchange data (database) | |
| EXCHANGE | Database descriptor is exchanged between neighbors | Type-2 DBD |
| LOADING | Contents of the LSA database is sent between neighbors (this is not a full table)<br>Content that is sent is based on the LSAs that the router misses and LSAs that are requested<br>LSR (Link State Request) requests an LSA from a neighbor<br>LSU (Link State Update) provides a neighbor with the LSA or LSAs<br>LSAck (Link State Acknowledgement ) confirms the reception of the LSU | Type-3 LSR<br>Type-4 LSU<br>Type-5 LSAck |
| FULL | Database is synced between neighbors | |

OSPF can be enabled on an interface using two methods:

- Network statement under router process (**network 0.0.0.0 0.0.0.0 area 0**)
- Interface command that enables ospf process (**ip ospf <PID> area 0**)
    - Interface configuration overrides network statement
    - Most specific network statement overrides other network statements

```
int gi0/0
 ip add 10.0.12.2 255.255.255.0
 ip ospf 1 area 21

router ospf 1
 network 10.0.12.0 0.0.255.255 area 0
 network 10.0.12.0 0.0.0.255 area 12
```

In the above config, the more specific network statement places interface gi0/0 in area 12

- However the interface statement > network statement so the interface is placed in area 21

## OSPF Packet Types

| | | |
|---|---|---|
| Type 1 | Hello Message | Discovers and monitors neighbors.<br>Sent periodically to 224.0.0.5 on all interfaces (link-local in scope).<br>Virtual-Links use unicast Hello packets.<br>On broadcast and NBMA networks, Hello packets are used to elect DR and BDR. |
| Type 2 | Database Descriptor (DD/DBD) | Synchronizes the link-state databases for all routers.<br>The Database descriptor is basically a summary of the OSPF LSA database<br>The routers only exchange the list of all LSAs they possess and update the ones that are missing from the database.<br>No actual LSAs are exchanged. |
| Type 3 | Request (LSR) | Requests for individual neighbors LSA details.<br>After DBD packets exchange process, the router may find it does not have an up-to-date database.<br>The LSR packet is used to request pieces of neighbor database that is more up-to-date or missing. |
| Type 4 | Update (LSU) | Response to LSR with LSA details. Implement the flooding of LSAs.<br>The local router advertises LSA with an LSU packet to its neighboring routers.<br>The local router also advertises the LSU packet with information in response to an LSR. |
| Type 5 | Acknowledgement (LSAck) | Confirmation of the reception of an LSU in response to an LSR. |

| **Stuck in** | **Reason** |
|---|---|
| WAIT | Long dead-interval |
| INIT | One way communication<br>Mismatched timers |
| 2-WAY | No DR elected<br>Possible stable state |
| EXSTART | MTU mismatch |
| EXCHANGE | MTU mismatch |
| LOADING | MTU mismatch |
| FULL | Possible stable state |

LSAs can be acknowledged using two methods:

- Sending a Link-State Acknowledgment (LSAck) message (**explicit acknowledgment**)

- Sending the same LSA that was received back to the other router in an LSU message (**implicit acknowledgment**)

## OSPF Adjacency Troubleshooting / Stuck States

**Stuck in WAIT-State**

- Reason: Unreasonably long dead-interval on broadcast and non-broadcast network type. This is because routers spend the dead-interval time (40 seconds by default) in the wait state before becoming FULL neighbors.
- Behavior: Routers will appear as DROTHERS even with priority set to non-zero value.
  show ip ospf interface: State will show as WAIT.
  show ip ospf neighbor: State will show as TWO-WAY.

**Stuck in INIT-State**

- Reason: One-way communication. OSPF not enabled on one side, L2 problem, mismatched Hello or Dead interval on ethernet interfaces. Dead interval can differ on P2P interfaces.
- Behavior: Routers will not show up as neighbors (blank), even though ospf has been enabled on both sides.
  show ip ospf interface: State will show DR for both routers, and correctly P2P on P2P interfaces.
  show ip ospf neighbor: No state.
- Troubleshoot local sent Hellos with show ip ospf interface, will show own router in INIT phase.
- Troubleshoot remote received Hellos with show ip ospf neighbor, if the neighbor is also shown in INIT phase then the problem is local.

**Stuck in TWO-WAY-State**

- Reason: Not always a problem (stable state for DROTHERS), problem when both routers on the same segment are set to priority 0 for a broadcast network. No DR is elected so no FULL neighborship can form.
  show ip ospf neighbor: State will show DROTHER for both routers.

**Stuck in EXSTART-State**

- Reason: MTU mismatch on both sides, this relates to unicast reachability. This indicates a unicast problem where multicast hellos are received properly, the master communicates to the slave that they should move on to exchange using unicast.
- Behavior: One side will show stuck in EXSTART (master) the other side will have moved on to EXCHANGE state (slave). The master will then tear down the adjacency after a number of failed retransmissions. After this the adjacency will restore and the process starts from the start.
  show ip ospf neighbor: State will show EXSTART for both master and EXCHANGE on slave.

**Stuck in EXCHANGE-State**

- Reason: Same as EXSTART.

**Stuck in LOADING-State**

- Reason: MTU mismatch but not between neighbors, but when intermediate device (switch) is configured with wrong MTU settings.
- Behavior: One router will show FULL state, the other will show LOADING state, or both will show LOADING state.
  show ip ospf interface: State will show DR for both routers, and correctly P2P on P2P interfaces.
  show ip ospf neighbor: No state.

**Stuck in FULL-State**

- Reason: No problem unless no routes are received. This indicates an OSPF database mismatch or some sort of route filtering configured on neighbors.

## OSPF Link-State Advertisement (LSA)

- LSA maximum age is 60min, refresh time is 30min.
- Check summing is performed on all LSAs every 10 minutes.
- The router keeps track of LSAs that it generates and LSAs that it receives from other routers.
- The router refreshes LSAs that it generated and it ages the LSAs that it received from other routers.
- Prior to the LSA group pacing feature, all LSAs would be refreshed and check summed at the same intervals.
- This process wasted CPU resources because only a small portion of the database needed to be refreshed.

## OSPF LSA Types

**Type 1: Router (All Routers)**

- Contents: Router ID, router interfaces & neighbors (not always!). Prefix information is not included.
- Originator: All OSPF routers generate a SINGLE Type-1 LSA. A router LSA can contain information about multiple links.
- Triggers an SPF recalculation.
- Flooding scope: Own area, information remains unchanged, not altered by others.

**Type 2: Network (DR)**

- Contents: Router IDs of all connected routers, netmask of subnets. (This is not included in Type-1 LSA in multi-access segments. Prefix information is not included.
- Originator: DR on the shared (broadcast, non-broadcast) segment. BDR in case of DR failure.
- Triggers an SPF recalculation.
- Flooding scope: Own Area, information remains unchanged, not altered by others.

**Type 3: Summary (ABR)**

- Contents: Calculated routing information for area routes,
- Based on the information from Type-1 and Type-2 LSAs. The must be in the routing table.
- Summarizes Type-1 and Type-2 LSAs, not the actual route prefixes.
- Summarizes other received Type 3 information.
- Originator: ABR sends this LSA into the area, summarizing Type-1 and Type-2 LSAs from other areas.
- Flooding scope: Own Area, information remains unchanged, not altered by others.

**Type 4: ASBR Summary (ABR)**

- Contents: Router ID of ASBR from another area, contains the route to the ASBR. The next hop of this route is updated to the RID of each ABR that forwards it.
- Originator: ABR of the area with an ASBR present.
- Flooding scope: Area, information is updated by each ABR.
- Renamed to Inter Area Router Link State in OSPFv3.

**Type 5: External (ASBR) / NSSA-ABR**

- Contents: Redistributed external routes. Includes the link cost to an external destination (E1 or E2)(E2 is default).
- Originator: ASBR, the next hop of the external route, remains the ASBR as is not updated by ABRs. Only the route towards the ASBR is updated using a Type-4 LSA.
- The NSSA ABR that translates Type-7 external LSAs into Type-5 also functions as an ASBR.
- Flooding scope: Entire domain (Except stub and NSSA areas).

**Type 7: NSSA External (ASBR)**

- Contents: Redistributed external routes from within an NSSA. These are translated into Type-5 by the NSSA ABR.
- Originator: ASBR in an NSSA area.
- Flooding scope: Own area (NSSA in which it was injected).

## IPv6 OSPF LSA Types

**Type 8: Link LSA (All Routers)**

- Contents: Link-local address and IPv6 prefix for the each link connected to the router.
- Originator: All OSPF routers.
- Flooding scope: Link-local.

**Type 9: Inter-Area Prefix LSA (All Routers)**

- Contents: IPv6 prefix or link-state changes.
- Originator: All OSPF routers.
- Flooding scope: Own area. The next hop of this prefix is updated to the RID of each ABR that forwards it.
- Basically replaces Type-3 LSA for each individual prefix.
- Does not trigger an SPF recalculation.

**Type 11: Grace LSA (All Routers)**

- Contents: Informing others that the router is undergoing a graceful restart.
- Originator: Restarting OSPF routers.
- Flooding scope: Link-local

# Filtering

## OSPF Filtering

- OSPF restricts most filtering to either ABRs or ASBRs, it is not possible to filter LSAs from within the same area
- It is always possible to filter prefixes between the OSPF database and the local RIB, even on internal routers
- Filtering options:
    - Filter Type-3 or Type-5 LSAs using **filter-lists** on ABR/ASBR
    - Filter routes using **distribute-lists** on all routers between the OSPF database and the local RIB

## OSPF Type-3 LSA Filtering using Filter-Lists

- Filter-lists only affect Type-3 LSAs and can only be used with prefix-lists
- Filters the entire LSA, not just the routing entry
- Can be configured in the inbound or the outbound direction

**R1 is ABR between area 0 and area 123, prevent route 192.168.0.3/32 from reaching area 0:**

```
ip prefix-list R3_Lo3 deny 192.168.0.3/32
ip prefix-list R3_Lo3 permit 0.0.0.0/0 le 32

router ospf 1
 area 0 filter-list prefix R3_Lo3 in
```

**Or:**

```
ip prefix-list R3_Lo3 deny 192.168.0.3/32
ip prefix-list R3_Lo3 permit 0.0.0.0/0 le 32

router ospf 1
 area 123 filter-list prefix R3_Lo3 out
```

## Filtering Type-3 LSAs using Summarization

- The **area <area-id> range** command can also be used to filter all routes that match the criteria
- This command only works on inter-area routes, not redistributed routes
- The **not-advertise** keyword will filter all the more specific routes and the summary, basically blocking the Type-3 LSAs
- Use the **summary-address** keyword for external routes

**R1 is ABR between area 0 and area 123, prevent internal routes 192.168.0.1-3/32 from reaching area 0:**

```
router ospf 1
 area 123 range 192.168.0.0 255.255.255.252 not-advertise
```

**R1 is ABR between area 0 and area 123, prevent external routes 192.168.0.1-3/32 from reaching area 0:**

```
router ospf 1
 area 123 summary-address 192.168.0.0 255.255.255.252 not-advertise
```

## Distribute-Lists

- Filtering routes using **distribute-lists** is only possible in the inbound direction
- Only one distribute list can be applied per process and can match an ACL, prefix-list or route-map
- It is not possible to filter LSAs from being received within the same area
- Distribute-Lists are a local filter of the LSA database and the RIB, they do not actually block the LSA from being received
    - Only prevents routes from being installed in the RIB

**Prevent R1 from installing the route to 192.168.0.3/32 in the local RIB:**

```
ip prefix-list R3_Lo3 deny 192.168.0.3/32
ip prefix-list R3_Lo3 permit 0.0.0.0/0 le 32

router ospf 1
 distribute-list prefix R3_Lo3 in
```

**Prevent R1 from installing any routes generated by R2:**

```
ip prefix-list R2 deny 10.0.12.2/32
ip prefix-list R2 permit 0.0.0.0/0 le 32
```

```
ip prefix-list PREFIXES permit 0.0.0.0/0 le 32

router ospf 1
 distribute-list prefix PREFIXES gateway R2 in
```

## Distribute-List Filtering from NSSA area

- Filtering from NSSA into normal areas is only needed on the router that performs the Type-7 translation (Highest RID).
- NSSA ABRs translate Type-7 LSAs into Type-5. The requirement is that the routes they translate exist in the routing table.
- Because distribute-lists filter routes from reaching the routing table, the function can be used to filter routes from both the NSSA ABR and all routers that exist in the backbone and other areas.

```
ip prefix-list Lo3 deny 192.168.0.3/32
ip prefix-list Lo3 permit 0.0.0.0/0 le 32

router ospf 1
 area 13 nssa
 distribute-list prefix Lo3 in
```

## Outgoing Database Filter

- Filter all outgoing LSAs on the specified interface.
- Filter all outgoing LSAs to the specified neighbor using the neighbor statement.

```
int fa0/0
 ip ospf database-filter all out

router ospf 1
 neighbor 10.0.12.2 database-filter all out
```

## Prefix-Suppression

- Only advertises prefixes associated with secondary IP addresses, and passive interfaces.
- This can be configured on a per interface basis or for the entire process.
- Basically, all primary addresses will be suppressed.
- Secondary IP addresses are only advertised by enabling OSPF on the interface, not the network statement.
- If prefix suppression was enabled for the entire process, only secondary addresses and loopbacks would be advertised.
- By specifying the secondaries none keyword, the secondary address is not advertised into OSPF.

```
router ospf 1
 prefix-suppression
 ip ospf 1 area 0 secondaries none
```

```
int lo0
 ip ospf prefix-suppression disable
```

# LFA / FRR

## Fast Reroute (FRR) Direct LSA

- IOS only supports per-link LFA.
- The high priority enables FRR for /32 prefixes only, the low priority enables FRR for all prefixes.
- The fast-reroute keep-all-paths option keeps all information in the table, including paths that were not chosen.
- When an area is specified, external routes are not a candidate for FRR. This is because they do not belong to an area.

```
router ospf 1
 fast-reroute per-prefix enable area 0 prefix-priority high
 fast-reroute per-prefix enable prefix-priority high
 fast-reroute keep-all-paths
```

**Configure a custom high prefix priority:**

```
ip prefix-list FRR permit 0.0.0.0/0 ge 30

route-map FRR permit 10
 match ip address prefix FRR

router ospf 1
 prefix-priority high route-map FRR
```

**Exclude interface in calculation:**

```
int fa0/0
 ip ospf fast-reroute per-prefix candidate disable
```

## FRR Tie Breakers

**On by default:**

- SRLG 10 - Shared Risk Link Group. Connected to same switch for example. Configure with srlg gid interface command.
- Primary Path 20 - Prefer backup path that is ECMP.
- Interface Disjoint 30 - Prefer backup path that exits through a different (sub)interface.
- Lowest-Metric 40 - Prefer backup path with the lowest metric.
- Linecard-disjoint 50 - Prefer backup path that exits through a different line-card
- Node protecting 60 - Prefer backup path that doesn't lead to the same router.
- Broadcast interface disjoint 70 - Prefer backup path that is in the same broadcast range.
- Load Sharing 256 - If no tie breakers, share backup paths in ECMP.

**Off by default**

- Downstream - Similar to feasibility condition.
- Secondary-Path - Prefer backup path that is not ECMP.

Manually specify tie breakers and index number (lower is more preferred).

- The required keyword forces matching. If no match, do not go to next-tie breaker and don't use the path.
- When manually configuring tie-breakers, others not included will not be used.

```
router ospf 1
 fast-reroute per-prefix tie-break lowest-metric required index 10
 fast-reroute per-prefix tie-break node-protecting required index 20
 fast-reroute per-prefix tie-break srlg required index 30

show ip ospf fast-reroute prefix
show ip route repair-paths
show ip ospf rib
```

# Misc

## OSPF TTL Security

- Normally OSPF packets are sent with a TTL of 1 or 2 for directly connected neighbors.
- With TTL Security the TTL for sent packets is set to 255 and received packets must match the configured value.
- Must be configured and on both sides.
- If TTL Security is configured with a hop count of 1, the router will only accept packets with a TTL of 254.

```
int fa0/0
 ip ospf ttl-security hops 1

router ospf 1
 ttl-security all-interfaces hops 1
int fa0/0
 ip ospf ttl-security disable
```

## OSPF Ignore MTU

- If two neighbors use different MTU settings on the link, the neighborship will not form.
- Override with the ip ospf mtu-ignore interface command.

```
int fa0/0
 ip ospf mtu-ignore
```

## Incremental SPF (iSPF)

- Faster convergence means that the routing protocol is more sensitive to oscillating processes, which in turn makes it less stable.
- iSPF keeps the SPT structure after the first SPF calculation and using it for further computation optimizations.

```
router ospf 1
 ispf
```

# OSPFv3 / IPv6

## OSPFv3 (IPv6 OSPF)

- Confusingly, Cisco refers to IPv6 OSPF as just OSPFv3 or 'Tradidional OSPFv3'
    - Whereas OSPFv3 that uses both IPv4 and IPv6 is referred to as 'OSPFv3 Address-Family Configuration'

## OSPFv3 LSA Differences

| **OSPFv2 Definition** | **OSPFv3 Definition** |
|---|---|
| Type-3 Summary LSA | Type-3 Inter-area Prefix LSA for ABRs |
| Type-4 ASBR-Summary LSA | Type-4 Inter-area Prefix LSA for ASBRs |

## OSPFv3 New LSAs

| **LSA Type** | **LSA Name** | **Definition** |
|---|---|---|
| Type-8 | Link LSA | Only exists on the local-link and only advertises IPv6 addresses<br>Advertises all IPv6 associated with the link (global / link-local)<br>Similar to Type-1 LSA, except limits the prefixes to a single link per LSA<br>Can set OSPFv3 specific bits that influence the networks advertised by the router<br>NU-Bit (No-Unicast) for example indicates that a network should not be used in calculations |
| Type-9 | Intra-area Prefix LSA | Sends information about IPv6 networks (including stub) attached to the router<br>Sends information about IPv6 transit network segments within the area<br>Basically this LSA combines OSPFv2 Type-1 and Type-2 LSAs into a single LSA for IPv6 networks |
| Type-10 / 11 | Opaque | Used as generic LSAs to allow easy future extension of OSPF<br>Type-10 has been adapted for MPLS traffic engineering |

```
ipv6 unicast-routing
ipv6 cef
!
ipv6 router ospf 1
 router-id 0.0.0.1
!
interface gi0/0
 ipv6 address fe80::1 link-local
 ipv6 ospf network point-to-point
 ipv6 ospf 1 area 0
 !
interface lo0
 ipv6 address fe80::1 link-local
 ipv6 address 1::1/128
 ipv6 ospf network point-to-point
 ipv6 ospf 1 area 0
```

## OSPFv3 Address Family (AF) Configuration

- Multiple instances of OSPFv3 AF can be run on a link
- OSPFv3 AF maintains a single Link-State database for all address-families
- An OSPFv3 AF process can be configured for the IPv4 and IPv6 address-family
- Not compatible with OSPFv2, neighborship will not form between an OSPFv2 and an OSPFv3 router

```
ipv6 unicast-routing
ipv6 cef
router ospfv3 1
 router-id 0.0.0.1
!
 address-family ipv6 unicast
  passive-interface default
  no passive-interface gi0/0
!
 address-family ipv4 unicast
  passive-interface default
  no passive-interface gi0/0
!
interface gi0/0
 ip address 10.0.12.1 255.255.255.0
 ipv6 address fe80::1 link-local
 ospfv3 1 ipv4 area 0
 ospfv3 1 ipv6 area 0
 ospfv3 network broadcast
!
interface lo0
 ip address 1.0.0.1 255.255.255.255
```

```
ipv6 address fe80::1 link-local
ipv6 address 1::1/128
ospfv3 network point-to-point
ospfv3 1 ipv4 area 0
ospfv3 1 ipv6 area 0
```

**Show commands**

```
show ospfv3 neighbors
show ospfv3 interface brief
show ospfv3 database
show ospfv3
```

# Session / Networks

## OSPF Network Types

| Name | Timers | Hello | Updates | Neighbor | DR / BDR | Mask | Default on type |
|---|---|---|---|---|---|---|---|
| Point-to-Point | 10 / 40 | Multicast | Multicast | No | No | - | Serial / Tunnel / FR P2P |
| Broadcast | 10 / 40 | Multicast | Multicast | No | Yes | - | Ethernet |
| Non-Broadcast | 30 / 120 | Unicast | Unicast | Yes | Yes | - | FR Physical / FR P2MP |
| Point-to-Multipoint | 30 / 120 | Multicast | Unicast | No | No | /32 | - |
| Point-to-Multipoint Non-Broadcast | 30 / 120 | Unicast | Unicast | Yes | No | /32 | - |
| Loopback | - | - | - | - | - | /32 | Loopbacks |

Neighbor statements are only accepted on non-broadcast network types
There is no need to specify an outgoing interface with the neighbor statement
OSPF will look in the RIB for the exit interface of the neighbor address and will verify that OSPF is configured as a non-broadcast network type

DR Networks

- 224.0.0.5 - All SPF-Routers multicast address
- 224.0.0.6 - All Designated Routers (DR) multicast address

- All LSA updates will be forwarded from the DROTHER routers to the DR/BDR using 224.0.0.6
- The DR will update the LSAs and propagate the changes to the rest of the DROTHER routers using 224.0.0.5

- The DROTHERS respond to the DR with a LSU or a LSAck using 224.0.0.6

## OSPF Database Networks (Link Types)

| Link Type | Description | LSID |
|---|---|---|
| 1 | Point-to-point connection to another router | Neighbor router ID |
| 2 | Connection to transit network | IP address of DR |
| 3 | Connection to stub network | IP Network |
| 4 | Virtual Link | Neighbor router ID |

OSPF Adjacency Process

Non-DR Networks

Hellos are sent to

Normally sent to 224.0.0.5 (All SPF Routers) and 224.0.0.6 (All Designated Routers).

| | |
|---|---|
| Point-to-Point | Hello sent to 224.0.0.5 |

## OSPF DR / BDR Election

- The DR/BDR election is not preemptive, the first router that boots will be selected.
- If all routers boot at the same time, the DR is chosen based on the highest priority, or the highest RID.
- The DR originates the network LSA on behalf of the network and forms adjacencies with all DROTHERS to synchronize the LSDB.
- BDR and DR perform the same function, except the BDR only sends out updates if the DR goes down.
- In NBMA networks the BDR is elected before the DR.

Primary DR Role

- All routing updates will be forwarded from the DROTHER routers to the DR/BDR using 224.0.0.6.
- The DR will update the LSAs and propagate the changes to the rest of the DROTHER routers using 224.0.0.5.

Secondary DR Role

- Informs all routers in the area of the shared segment using Type-2 LSA.
- This is only performed by the DR, not the BDR. The BDR only receives LSAs and is promoted if the DR fails.

# Summarization

## OSPF Inter-Area Route Summarization (ABR/ASBR)

- Routes can only be summarized manually at the area boundaries (ABR) or at the ASBR
- The **area range** command can summarize inter-area routes for which it is about to create a Type-3 Summary LSA
- The area that is specified is the area where the routes are located, not the area that is being summarized into
    - ABR does not advertise the subordinate route Type-3 LSAs
    - If the subordinate route does not exist, the summary will not be created

```
router ospf 1
 area 1 range 172.16.0.0 255.255.0.0
```

- The metric (cost) chosen for the summary is the cost associated with the best subordinate route
    - It is also possible to associate a new (lower) cost value with the summary by using the **area <area id> range <prefix> cost** keyword
- Basically, the more specific Type-3 LSAs are suppressed and instead a single Type-3 LSA is advertised to the adjacent area

**Summarize 3 prefixes with different cost:**
192.168.0.1/32 cost 2
192.168.0.2/32 cost 3
192.168.0.3/32 cost 4

A summary of 192.168.0.0/30 will receive of cost of 2

## OSPF External Route Summarization (ASBR)

- The **summary-address** command can only summarize Type-5 External LSAs or Type-7 LSAs that it is about to translate to a Type-5 LSA
- This can only be performed on an ASBR (usually the router that redistributes the routes)
- In the case of NSSA, the ABR to the NSSA will also be considered an ASBR and will be able to further summarize external routes
- The **nssa-only** keyword will keep the summary (and the more specific routes) inside the NSSA
- Configure the **nssa-only** keyword on the redistributing ASBR inside the NSSA area, not the ABR that connects to the NSSA area

```
router ospf 1
 summary-address 172.16.0.0 255.255.0.0 nssa-only
```

- The metric (cost) chosen for the summary is the cost associated with the best subordinate route
    - It is NOT possible to associate a new cost value with the summary

## OSPF Discard Route

- The discard route is installed automatically when creating a summary and points to the null0 interface
    - The AD for inter-area discard route (created with **area-range**) is 110
    - The AD for external discard route (created with **summary-address**) is 254
- Disable the discard route with the **no discard-route <internal/external>** command

```
router ospf 1
 no discard-route internal
 no discard-route external
```

## OSPF Default Route

- It is possible to generate a default route (and suppress all subordinate routes) with the **area-range** or **summary-address** command

**Only advertise a default route from area 0 into other areas**

```
router ospf 1
 area 0 range 0.0.0.0 0.0.0.0
```

- Another method is to use the **default-information originate** command
- The originating router will create a Type-5 External LSA for an external E2 0.0.0.0/0 prefix
- The originating router will become an ASBR even if it doesn't redistribute any other routes
    - The Type-5 LSA will only be created if the router has a route to 0.0.0.0/0 from another protocol in its routing table
    - The **always** keyword will force the route to be always advertised, regardless if the 0.0.0.0/0 prefix exists in the routing table
- The default route will be originated as a Type-2 External (E2) route
- The **default-information originate** statement does NOT require the presence of a default route when configured on an NSSA ABR

**OSPF Conditional Default Route Injection**

```
int se0/0
 ip address 1.0.0.1 255.255.255.252

ip route 0.0.0.0 0.0.0.0 1.0.0.2

ip prefix-list UPLINK_NETWORK permit 1.0.0.0/30

route-map DEFAULT_RM permit 10
 match ip address prefix-list UPLINK_NETWORK

router ospf 1
 default-information originate metric 5 metric-type 2 route-map DEFAULT_RM
```

- The above configuration will only advertise the Type-5 LSA into OSPF if the 1.0.0.0/30 network towards the ISP is operational (link is up)

- If the link goes down between the ASBR and the ISP router, the 1.0.0.0/30 will be withdrawn from the RIB, and thus the default route from OSPF

# Stub Area Types

## OSPF Area Types

| Type | Underlying Area | LSA Allowed | 0.0.0.0/0 route inserted by default | Default route type | Default route generated by |
|---|---|---|---|---|---|
| NORMAL | NORMAL | 1,2,3,4,5 | No | Type-5 | default-information-originate [always] |
| STUBBY | STUB | 1,2,3 | Yes | Type-3 | area 1 stub |
| T-STUBBY | STUB | 1,2,[3] | Yes | Type-3 | area 1 stub no-summary |
| NSSA | NSSA | 1,2,3,7 | No | Type-7 | area 1 nssa default-information-originate |
| T-NSSA | NSSA | 1,2,[3],7 | Yes | Type-3<br>Type-7 | area 1 nssa no-summary<br>area 1 nssa no-summary default-information-originate |

## OSPF Stubby Area

- ABRs do not flood Type-5 LSAs into the STUBBY area
- Internal STUBBY routers ignore received Type-5 LSAs by ABR, does not insert type Type-4 and Type-5 LSA
- Default route is installed automatically into STUBBY and T-STUBBY areas with a cost of 1
- OSPF adjacency rules dictate that neighbors need to use the same area type
    - If a router is configured for stub, the neighboring router in the same area also needs to be a stub

## OSPF Totally Stubby (T-STUBBY) Area

- Totally Stubby area is a stubby area with additional filtering
- T-STUBBY also stops prevents ABRs from flooding Type-3 LSAs, except for a default route
- Internal routers in T-STUBBY area have no knowledge of external/inter-area prefixes besides default route that is converted to Type-3
- Only the ABRs need to be configured with **no-summary** keyword, configuring internal routers as TSTUBBY will not cause issues however

```
router ospf 1
 area 123 stub no-summary
```

## OSPF NSSA Areas

- ABRs do not flood Type-5 LSAs into the NSSA area
- Internal NSSA routers ignore received Type-5 LSAs by ABR, does not insert type Type-4 and Type-5 LSA
- NSSA is a NSSA area type that allows creation of Type-7 LSA that injects external routes into other areas
- Default route is not installed automatically NSSA areas
- Specify **no-redistribution** on the NSSA ABR to have local redistributed routes only be redistributed into normal areas, not the NSSA

- Redistributed external routes in NSSA area will be shown as N1 or N2 type routes
    - These routes will be converted to regular E1 and E2 when the ABR translates those routes into other normal areas
    - When using multiple NSSA ABRs, the router with the highest RID is responsible for translating Type-7 LSAs into the normal area
    - The forward-address is set on the translated prefix in order to allow for ECMP

Totally Stubby NSSA (T-NSSA)

- Totally Stubby NSSA area is a NSSA area with additional filtering
- T-NSSA also stops prevents ABRs from flooding Type-3 LSAs, except for a default route
- Only the ABRs need to be configured with the **no-summary** keyword, configuring internal routers as T-NSSA will not cause issues however

```
router ospf 1
 area 123 nssa no-summary no-redistribution
 area 123 nssa default-information-originate metric 1 metric-value metric-type 2 nssa-only
 area 123 default-cost 1
```

The **default-cost** keyword will affect the cost of the redistributed default-route. Not other routes
NSSA **default-information-originate** does not require the presence of a default route when configured on the NSSA ABR

- Originating the default route on routers within the NSSA the presence of a default route is required
- The **nssa-only** keyword will limit the default route to the NSSA area only by setting the propagate (P) bit in the type-7 LSA to zero
- The F-bit indicates that a forwarding address is included in the LSA when set (Used to forward Type 7 LSA)

## OSPF NSSA Forward Address

- Always inserted into Type-7 LSA (interface IP-address) loopback has preference.
- Translated by default into Type-5 LSA with forward address intact.
- Based on this forward address, routers outside the NSSA will choose the best path towards the NSSA ASBR.

```
router ospf 1
 area 123 nssa translate type7 suppress-fa
```

The suppress-fa keyword will stop the forwarding address of the Type-7 LSAs from being placed in the Type-5 LSAs.

- This keyword takes effect only on an NSSA ABR or an NSSA ASBR.
- The P-bit is used in order to tell the NSSA ABR whether to translate type 7 into type 5. P=1 means translate.
- When using multiple NSSA ABRs, if suppression is enabled on the translating NSSA ABR (highest RID) ECMP will stop functioning.
- In this case the translating router (highest RID) will become the next-hop, because the forward address is not known (0.0.0.0).

# VL / GRE

## OSPF Virtual-Links

- Only work over normal (transit) areas and do not operate over stub areas and NSSA areas
- Always in area 0, even if they are configured on other areas. Use area 0 authentication by default
- Endpoints are the RIDs, not actual ip-addresses
- Can only cross one area
- P2P in nature and unnumbered, and they carry only OSPF communication such as hellos and LSAs
- Existing VLs can easily be spotted by the DNA bit set in the OSPF database, VLs also set the V-bit to 1
- Default hello-timer is 10 seconds and dead timer is 40 seconds
- The ttl-security hops keyword specifies over how many hops the Virtual-Link is allowed to travel
- The area that you specify with **area …. virtual-link** command is the area the VL travels over

```
router ospf 1
 area 23 virtual-link 192.168.0.3 ttl-security hops 2
```

```
show ip ospf virtual-link
show ip ospf database
```

## OSPF GRE

- Beware of recursive routing when choosing tunnel endpoints.
- Tunnel between ABRs, not the routers that should be linked in the same area.

```
int fa0/0
 ip ospf 1 area 1234
int fa0/1
 ip ospf 1 area 0
int tun0
 ip unnumbered fa0/1
 tunnel source fa0/1
 tunnel destination 10.0.23.2
 tunnel mode gre ip
 ip ospf 1 area 1234
```

# Quality of Service (QoS)

## QoS

- Simplest form of traffic queing is tail drop, which simply drops new packets when the queue os full
- This can lead to problems in TCP traffic and cause 'global synchronization'
- All hosts simultaneously reduce their transmission rate when a packet is dropped, afterwards they will all resume simulatiously
- TCP starvation / UDP dominance is caused when UDP and TCP traffic are mixed, the TCP hosts will throttle but UDP will keep using bandwidth
- This can happen when there is no QoS, or tcp/udp is combined in the same class-map, this can not always be avoided however

## TCP

- A TCP window is the number of data bytes that the sender is allowed to send before waiting for an acknowledgment, the higher the more performance
- The window size is defined by receiver, the defualt size is 4128 bytes (33024 bits), can be modified with the **ip tcp window-size** command
    - The remote host/receiver will also need to support (and be configured for) receiving this larger window size
- TCP selective acknowledgment improves perforamnce if multiple packets in a window are lost
    - The receiver notifies the sender (acknowledgments) that data has been received
    - Enable with the **ip tcp selective-ack** global command
    - Only used when multiple packets are lost in a window, not during 'regular' tcp operations
    - The feature can be used alongside:
        - TCP ECN (Explicit congestion notification). Router suggests host to slow down traffic, enable with **ip tcp ecn** command
        - TCP Keepalives. Identifies dead connections
        - TCP time stamps. Improves round-trip measurements, enable with **ip tcp timestamp** command

```
ip tcp ecn
ip tcp selective-ack
ip tcp timestamp
```

## VoIP

Different applications require different treatment, the most important parameters are:

- Delay: The time it takes from the sending endpoint to reach the receiving endpoint.
- Jitter: The variation in end to end delay between sequential packets.
- Packet loss: The number of packets sent compared to the number of received as a percentage.

One-way requirements for voice:

- Latency $\leq$ 150 ms (Delay)
- Jitter $\leq$ 30 ms

- Loss ≤ 1%
- Bandwidth (30-128Kbps)

Hardware Queue

- Hardware transmit (Tx) queue is FIFO by default for ethernet interfaces. Tx queue is 256.
- Hardware queue is WFQ by default for serial interfaces. Tx queue is 64.

```
int fa0/0
 tx-ring-limit 256

show controllers fa0/0 | i tx
```

## Class of Service (CoS)

- 802.1p is L2 information.
- ToS / DSCP is L3 information, it will remain constant between endpoints.

802.1p

| | | |
|---|---|---|
| 111 | 7 | Reserved |
| 110 | 6 | Reserved |
| 101 | 5 | Voice |
| 100 | 4 | Video |
| 011 | 3 | Voice Signal |
| 010 | 2 | High Data |
| 001 | 1 | Low Data |
| 000 | 0 | Best Effort |

## L3 Type of Service (ToS)

- ToS / DSCP uses left 3 bits with IPP, uses left 6 bits with DSCP and 8 bits with ToS.
- Drop Probability (DP) right-most bit is always 0.
- Flow Control (FC) is always 0 unless ECN is used.
- IP Precedence (IPP) part of DSCP is called the Per-Hop Behavior (PHB).

| Class | Binary | DSCP | ToS |
|---|---|---|---|
| cs1 | 001\|000\|00 | 8 | 32 |
| af11 | 001\|010\|00 | 10 | 40 |
| af12 | 001\|100\|00 | 12 | 48 |
| af13 | 001\|110\|00 | 14 | 56 |

| | | | |
|---|---|---|---|
| cs2 | 010\|000\|00 | 16 | 64 |
| af21 | 010\|010\|00 | 18 | 72 |
| af22 | 010\|100\|00 | 20 | 80 |
| af23 | 010\|110\|00 | 22 | 88 |
| cs3 | 011\|000\|00 | 24 | 96 |
| af31 | 011\|010\|00 | 26 | 104 |
| af32 | 011\|100\|00 | 28 | 112 |
| af33 | 011\|110\|00 | 30 | 120 |
| cs4 | 100\|000\|00 | 32 | 128 |
| af41 | 100\|010\|00 | 34 | 136 |
| af42 | 100\|100\|00 | 36 | 144 |
| af43 | 100\|110\|00 | 38 | 152 |
| cs5 | 101\|000\|00 | 40 | 160 |
| ef | 101\|110\|00 | 46 | 184 |
| cs6 | 110\|000\|00 | 48 | 192 |
| cs7 | 111\|000\|00 | 56 | 224 |

# Maps

## Class-Maps

- With match-all, all criteria must be met in order to have a match. Default.
- With match-any, only one of the criteria has to be met in order to have a match.
- The ip precedence and ip dscp keyword only match on IPv4 traffic.

**Match telnet traffic that is not marked with the default value of cs6:**

```
class-map match-all TELNET
 match protocol telnet
 match not dscp cs6

policy-map TELNET
 class TELNET
  drop

int fa0/0
 service-policy input TELNET
```

## Hierarchical Policy-Map

1. Police traffic matching the ACL to 64000 bps.
2. Police traffic matching a subset of this ACL with IPP0, IPP1, IPP2 to 32000 bps.
3. Police traffic matching a subset of this ACL with IPP2 to 16000 bps.
4. Always nest the most specific subset into the upper level policy-map.
5. In this case IPP2 with 8000 bps is the most specific, so this will be the lowest level policy-map.

```
ip access-list standard QOS_TRAFFIC
 permit 10.10.10.0 0.0.0.255

class-map match-all LEVEL_1_CM
 match access-group name QOS_TRAFFIC
class-map match-all LEVEL_2_CM
 match precedence 0 1 2
 match access-group name QOS_TRAFFIC
class-map match-all LEVEL_3_CM
 match precedence 2
 match access-group name QOS_TRAFFIC

policy-map LEVEL_3_PM
 class LEVEL_3_CM
 police 16000
policy-map LEVEL_2_PM
 class LEVEL_2_CM
 police 32000
 service-policy LEVEL_3_PM
policy-map LEVEL_1_PM
 class LEVEL_1_CM
 police 64000
 service-policy LEVEL_2_PM

int fa0/0
 service-policy output LEVEL_1_PM
```

# NBAR

## NBAR

- NBAR uses deep packet inspection instead of just matching on the specified port.
- This is more CPU-intensive than matching with an ACL.
- Managing with an ACL should be used if a previous devices has already performed deep-packet inspection with NBAR.
- List all known ports with show ip nbar port-map.
- Map a well-known port of a protocol to a new port with ip nbar port-map http 80 8080.

## NBAR Protocol-Discovery

- Monitor traffic protocols known to NBAR on a specific interface. CPU intensive.

```
int fa0/0
 ip nbar protocol-discovery

show ip nbar protocol-discovery
```

# Policing / Shaping

## Policy-Maps

- Policy-maps can be applied in both the ingress and egress direction.
- CBWFQ and LLQ policy-maps can only be applied in the egress direction.
- Shaping should be applied in the egress direction. But can be applied ingress.
- Policing should be applied in the ingress direction. But can be applied egress.

## Terminology

- Access rate (AR). This is the actual speed of the physical port.
- Committed Information Rate (CIR). Average rate the shaper is targeting in bps.
- Time Committed (Tc). Time interval in ms to emit traffic bursts.
- Burst Committed (Bc). Amount of bits that should be sent every Tc.
- Burst Excessive (Be). Amount of bits exceeding Bc that could be sent during Tc. Accumulated by idle periods.

## Policing

- Can be used to drop incoming packets that do not conform to the policy.
- Cisco routers have a Tc default value of 125 ms = 8 times a second.
- If the goal is 50Mbit/sec transmit speed on a 100Mbit/sec interface. The CIR would be specified as 50Mbit.
- The IOS will automatically calculate the Bc based on the configured CIR (recommended).
- In order to calculate the Bc the CIR needs to be divided by 8 or 4 (depending on the platform).
- Conform. Traffic is under Bc.
- Exceed. Burst size exceeds Bc, but under Bc+Be.
- Violate. Burst size exceeds Bc+Be.

Tc = CIR/Bc
Bc = CIR/Tc

CIR = 50 Mbit/sec = 50000000 bits/sec
Bc = 50000000 / 8 = 6250000 bits / 8 = 781250 bytes

```
policy-map POLICER
class class-default
```

```
 police cir 50000000 bc 781250
  conform-action transmit
  exceed-action drop

int fa0/0
 service-policy input POLICER
```

## Shaping

- Shaping is applied by altering the time in which traffic is allowed to send, not the speed of the port.
- This means that the average traffic sent will be less over time.
- The IOS will automatically calculate the Bc and Be based on the configured CIR (recommended).
- With a shape average (CIR) of 50 Mbit, the calculated Bc and Be will be 200000.
- The Tc will be 250, meaning that every 4ms (1 second / 250) the interface will forward at the line speed.
- The shaper can also be based on a percentage of the link speed, however this is dependent on the manually configured bandwidth, not the hardcoded line speed.

Tc = CIR/Bc
Bc = CIR/Tc

CIR = 50 Mbit/sec = 50000000 bits/sec
Bc = 50000000 / 200000 = 250

```
policy-map SHAPER
 class class-default
  shape average 50000000 200000 200000

int fa0/0
 service-policy output SHAPER
```

# Pre-Classify

## QoS Pre-Classification

- By default tunneling and VPN operations are applied before the QoS policy. QoS pre-classify (PQ) reverses the order.
- PQ on crypto map affects all tunnels on that physical interface.
- PQ on tunnel affects only that specific tunnel interface.
- PQ is needed when classification is based on IP address, ports, etc. Or a crypto map is used.
- PQ is not needed when classification is based on ToS. Or a tunnel interface is used.
- Can be enabled regardless, very little impact on performance.

```
interface tun0
 qos pre-classify
```

```
crypto map CMAP 10 ipsec-isakmp
 qos pre-classify
```

# Rate-Limiting

## Rate-Limiting

- Requires CEF. Can be configured on physical or sub-interface.
- Works similar to policer without MQC configuration.
- Like a policer, the CIR is configured in bits. The Bc and Be are configured in bytes.
- Match on all traffic or specific DHCP, QoS group, ACL (no named ACL support).

```
int fa0/0
 rate-limit input 50000000 781250 781250 conform-action transmit exceed-action drop

show interfaces rate-limit
```

# WRED

## Weighted Random Early Detection (WRED)

- Only works for TCP traffic. Enable in the egress direction.
- Drop preference values (AF) are used by WRED.
- WRED turns any queue on the interface into FIFO. The minimum threshold is the FIFO's queue depth before WRED is activated.
- The overall size of the queue depth is specified by the hold-queue command.
- The hold-queue must be higher than the minimum threshold configured in WRED.

```
int fa0/0
 hold-queue 40 out

show interface | i queue
```

The default WRED values differ per precedence and DSCP values.

- Minimum-threshold. Above this threshold WRED engages and starts randomly dropping packets.
- Maximum-threshold. Above this threshold TAIL-DROP engages and starts dropping packets (WRED is basically disabled).
- Mark-probability. Amount of packets dropped up until maximum threshold is reached (1 out of 10, 1 out of 5, etc).
- The Exponential Weighting Constant (EWC) alters how quick WRED reacts.
- Higher EWC value makes WRED react more slowly, lower EWC value makes WRED react more quickly (default is 9).

**Change IPP 0 to 15 min, 30 max and 1/5:**

```
policy-map WRED_IPP
 class class-default
  random-detect precedence-based
  random-detect precedence 0 15 30 5
 random-detect exponential-weighting-constant 9

int fa0/0
 service-policy WRED_IPP out
```

**Change DSCP 46 (EF) to 35 min, 40 max, and 1/2:**

```
policy-map WRED_DSCP
 class class-default
  random-detect dscp-based
  random-detect dscp 46 35 40 2
  random-detect exponential-weighting-constant 9

int fa0/0
 service-policy WRED_DSCP out
```

## AF Values

- AF11 is IP Precedence 1 (cs1) with low drop preference.
- AF23 is IP Precedence 2 (cs2) with high drop preference.
- AF32 is IP Precedence 3 (cs3) with medium drop preference.

| Drop Chance | Class #1 | Class #2 | Class #3 | Class #4 |
|---|---|---|---|---|
| Low | (AF11) 001010 | (AF21) 010010 | (AF31) 011010 | (AF41) 100010 |
| Medium | (AF12) 001100 | (AF22) 010100 | (AF32) 011100 | (AF42) 100100 |
| High | (AF13) 001110 | (AF23) 010110 | (AF33) 011110 | (AF43) 100110 |

## Explicit Congestion Notification (ECN)

- Mode of WRED that can be enabled to suggests a traffic flow to slow down, instead of actually dropping packets.
- Uses the last two bits of ToS (Flow Control value). Instead of dropped a packet WRED sets both these bits (ECT and CE) to 1.
- When the destination receives a TCP packet with both ECT and CE set to 1, it sets the ECE (Explicit Congestion Experienced) flag on its next TCP packet back to the sender.
    - When the sender receives the packet, it is instructed to slow down.

```
policy-map WRED_ECN
 class class-default
  random-detect ecn

int fa0/0
 service-policy WRED_ECN out
```

ECN Values

- The 7th bit is the ECT bit and the 8th bit is the CE bit.
- If a TCP host supports ECN, it sets either (but not both) of the low-order bits in the DSCP byte - ECT or CE - to 1.
- If a TCP host doesn't support ECN, these will both be set to 0.

| | |
|---|---|
| 00 | Not ECN capable |
| 01 | Endpoints are ECN capable |
| 10 | Endpoints are ECN capable |
| 11 | Congestion experienced |

# RIP

## RIP Passive Interfaces

- Passive interfaces transmit no protocol-related data, but still receive data
  - Protocol related data in case of RIP is only the updates
  - This includes RIP updates that cannot be sent, but still be received
  - This is because RIP does not use a hello mechanism to maintain adjacencies
- Updates can still be exchanged over passive-interfaces by using the **neighbor** statement

```
router rip
 passive-interface default
 neighbor 10.0.12.1
```

## RIP Redistribution

- Routing protocols or static routes redistributed into RIP require a set metric
- With a transparent the redistributed route will inherit the metric that is seen in the routing table of the redistributing router
- Use the transparent keyword only when you know that the metric is lower than 16.

- Redistributed OSPF E2 routes (cost 20) will always lead to an infinite metric when using the transparent keyword

```
router rip
 redistribute ospf 1
 default-metric 5
```

## RIP Summarization

- RIP does not install a discard route by default. Instead it relies on route poisoning when a routers own summary is received
- Manually add a discard route to null0 with ip route 0.0.0.0 0.0.0.0 null0
- RIP will generate a default route with the default-route originate command whether it exists or not in the routing table

Originate a default-route out of a specific interface (may lead to routing loops)

```
route-map RIP permit 10
 set interface se0/0

router rip
 default-information originate route-map RIP
```

## Validate Update Source

- Ensures that the source IP address of incoming routing updates is on the same IP network (enabled by default)
- Disabling split horizon on the incoming interface will also cause the system to perform this validation check
- Disable when using PPP IPCP addressing
- For unnumbered IP interfaces no checking is performed

```
router rip
 validate-update-source
```

## RIP Split Horizon

- Does not advertise the same networks out of interfaces from which they are learned (on by default except on FR and ATM)
- Poison Reverse is a 'stronger' variant of this
  - The routes are advertised out of the interfaces, but with an unreachable metric
- Poisoned Reverse overrides Split Horizon
  - Not implemented in Cisco RIPv2
  - Not enabled by default in RIPng

# RIPv1 / RIPv2

## RIPv1

- RIPv1 updates are sent to 255.255.255.255 using UDP port 520 by default
- RIPv1 does not support classless routing updates
- RIPv1 will always use automatic summarization

```
interface fa0/0
 ip address 10.0.12.1 255.255.255.252
 no shutdown
 !
interface lo0
 ip add 1.1.1.1 255.255.255.255
!
router rip
 network 10.0.0.0
 network 1.0.0.0
 no auto-summary
```

**show ip protocols**

```
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 24 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface             Send  Recv  Triggered RIP  Key-chain
    FastEthernet0/0       1     1 2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.0.12.1             120      00:00:22
  Distance: (default is 120)
```

**show ip route rip**

```
R     1.0.0.0/8 [120/1] via 10.0.12.1, 00:00:23, FastEthernet0/0
```

```
show ip rip database
1.0.0.0/8    auto-summary
1.0.0.0/8
    [1] via 10.0.12.1, 00:00:16, FastEthernet0/0
10.0.0.0/8    auto-summary
10.0.12.0/30    directly connected, FastEthernet0/0
```

- When configuring RIPv1 only, the router will send v1 and receive v1/v2
- Using the command **version 2** in router configuration mode will hardcode all ports to send and receive v2 only
    - This is not backwards compatible
    - RIPv1 is forwards compatible with routers running v2 only, because it also listens for v2 updates on 224.0.0.9 by default
    - Override router configuration with **ip rip send/receive version 1 2** on the interfaces
    - Interface configuration overrides router configuration

```
interface fa0/0
 ip rip send version 1 2
 ip rip receive version 1 2
```

### RIPv2

- RIPv2 updates are sent to 224.0.0.9 using UDP port 520 by default
- Send v2 updates to the broadcast address using the ip rip v2-broadcast interface command
    - This does not mean the updates will be accepted by clients running RIPv1 only

# Authentication

### RIP Authentication

- RIPv1 does not support authentication
- RIPv2 supports MD5 authentication and plain-text
    - The default authentication mode for RIPv2 is plain-text, even if not specifically configured

```
key chain RIP
 key 1
  key-string cisco

int fa0/0
 ip rip authentication mode md5
```

```
ip rip authentication key-chain RIP
```

# Filtering

## RIP Filtering

- Filtering can be done with distribute-lists or offset-lists
- Offset lists manipulate the RIP metric to an infinite value
- Distribute lists filter based on access- or prefix-lists

Filter all routes from se1/0 (R2)

```
ip prefix-list PREFIX permit 0.0.0.0/0 le 32
ip prefix-list NOT_R2 deny 10.0.12.2/32
ip prefix-list NOT_R2 permit 0.0.0.0/0 le 32

router rip
 distribute-list prefix PREFIX gateway NOT_R2 in se1/0
 offset-list 0 in 16 se1/0
```

# RIPng

## RIPng

- Passive interfaces are not supported
- Static (manual) neighbors cannot be configured (no neighbor command)
- Split Horizon and Poison Reverse can be activated only on a per-process basis, not on individual interfaces
- RIPng Updates are send to FF02::9 using UDP port 521 by default (RIPv2 uses UDP port 520)

```
ipv6 router rip RIPng
 timers 30 180 0 120
 maximum-paths 16
 distance 120
 split-horizon
 poison-reverse
 port 521 multicast-group FF02::9

show ipv6 rip
show ipv6 rip database
show ipv6 rip next-hops
```

## RIPng Summarization

- The default-information originate command will originate a default route in addition to all other specific routes
- The default-information only command will originate a default route and suppress all other specific routes
- Like RIPv2, the default-route does not have to exist in order to be advertised
- When the default-route is advertised, the router will ignore reception of all other default routes received on any interface
- The default metric of the default-route is 1
- The summary address copies the metric of the more specific routes it is summarizing

```
int fa0/0
 ipv6 rip RIPng enable
 ipv6 rip RIPng default-information originate metric 1
 ipv6 rip RIPng summary-address 2001::/16
```

## RIPng Metric

- The metric is incremented by the receiver instead of the sender
- Offset lists are configured only on interfaces and are for all received subnets, use the metric-offset command
- The metric-offset specified replaces the original increment of the metric (1)
    - Meaning that a metric-offset with value 4 will increase the metric by adding 3 to the existing value (4-1)

```
int fa0/0
 ipv6 rip RIPng enable
 ipv6 rip RIPng metric-offset 4
```

## RIPng Redistribution

- Routes redistributed into RIPng do not require a set metric
- Default behavior is to redistribute with a transparent metric
- This will inherit the metric from the other protocol
- Can lead to infinite metric (16)

Default behavior is to not redistribute connected interfaces  for the other routing protocol

- Override with the **include-connected** statement
- This advertisement can be overruled with a route-map filter

# Updates & Timers

## RIP Timers

| Type | Interval | Purpose |
|---|---|---|
| UPDATE | 30s | Exchanges routes |
| INVALID | 180s | Reset when UPDATE is received<br>Declares route invalid after timer expires |
| HOLD-DOWN | 180s | Starts after INVALID timer has expired<br>Marks route as unreachable |
| FLUSH | 240s | Reset when UPDATE is received<br>Removes route after timer expires |
| SLEEP | Disabled | Interval that postpones routing updates<br>in the event of a FLASH UPDATE |

- Because the flushed after timer expires after 240 seconds, the effective hold-down period is only 60 seconds

## RIP Updates

RIP sends routing update based on two conditions:

- Request message (flash update) is sent immediately when there is a change to the topology
- Full update is sent when RIP is started, at the regular interval or when routes are cleared from the RIB
    - Full updates contains learned and connected RIP routes.

- RIP Route Poisoning advertises a truly unreachable route to quickly flush it from routing tables

## RIP Flash-Update-Threshold

- If there is a change in the topology at 27 seconds, RIP will send a Flash Update, and then again a full update at the 30 second interval
- So in 3 seconds 2 updates are send flooding the network
- The flash-update-threshold can be configured to stop this behavior and will delay flash updates if they are within the interval

```
router rip
 flash-update-threshold 10
```

## Triggered Updates

When triggered extensions to RIP are enabled, updates are sent on the WAN only if one of the following events occurs:

- The router receives a specific request for a routing update. (Full database is sent)
- Information from another interface modifies the routing database. (Only latest changes are sent)
- The interface comes up or goes down. (Partial database is sent)
- The router is first powered on, to ensure that at least one update is sent. (Full database is sent)

```
int se1/0
 ip rip triggered
```

# Spanning Tree

## Spanning Tree Protocol (STP) Workings (802.1D)

- The function of STP is to prevent bridging loops
  - These exist because switches are not aware of other switches on the network and will loop frames
- STP works by establishing a hello mechanism between two neighboring switches (bridges in STP)
  - These hello's are called Bridge Protocol Data Units (BPDUs) and
  - BDPUs are sent to the L2 multicast address 0180.c200.0000
  - 802.1D (STP) frames are sent in the native VLAN
  - (R)PVST+ will also sent BPDUs to 0100.0ccc.cccd
- The entire goal of STP is to elect a root bridge and establish a reference point

When switches first come online they will start sending BPDUs on all active links

- The goal is to detect other switches on the segment and decide on a winner for each segment
- A segment is just a link between two switches
- The winner is the switch that has the lowest bridge ID (see STP Root Election below)

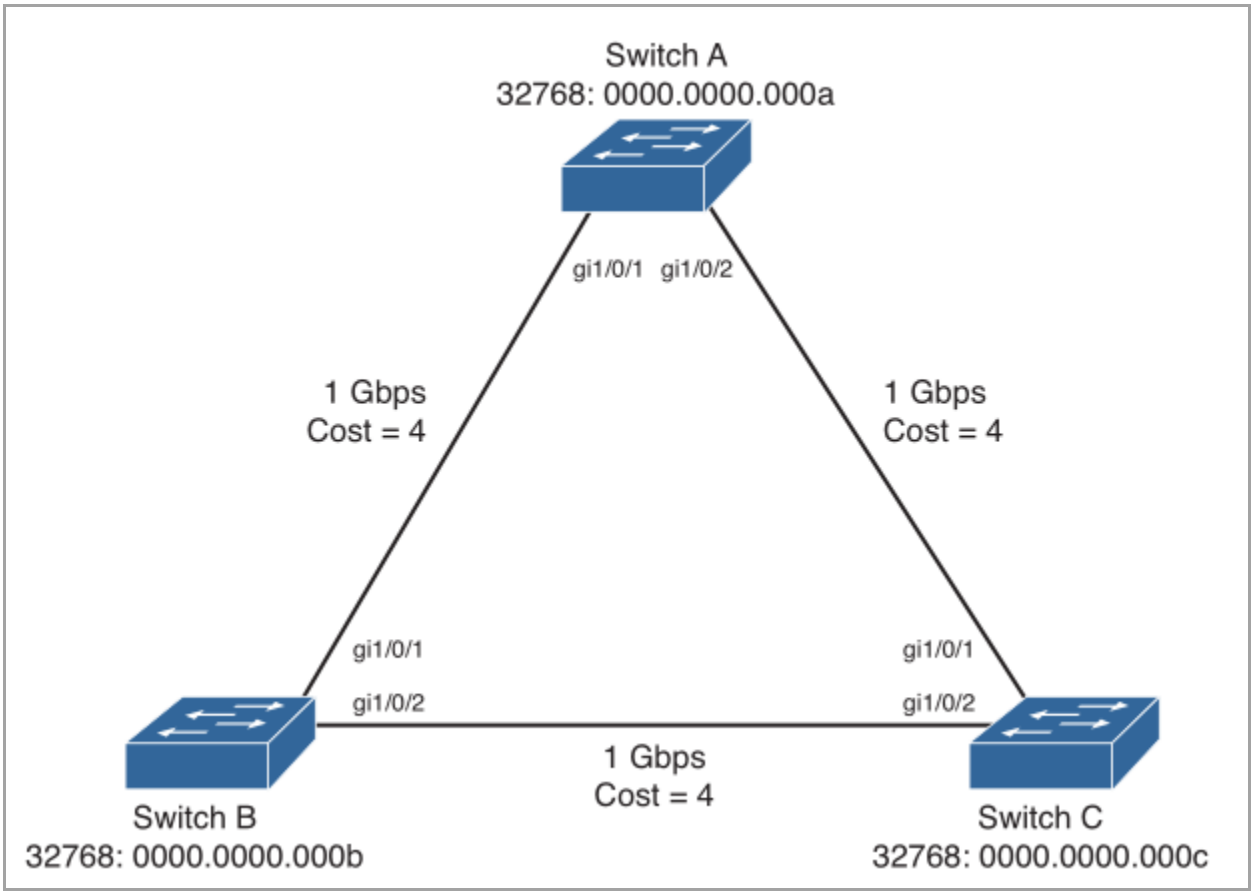
When the network converges there will be 1 overall winner (the root bridge) which will become the center of STP

- The root bridge will be the only one sending the BPDUs after convergence
- All other bridges will forward these BPDUs and will add their costs to the root bridge to the BPDU
- Between non-root bridges, election still happens to decide which bridge is the winner for each individual segment
  - The loser of this election will put its port in the blocking state, the winner will put its port in the forwarding state

## STP Bridge Protocol Data Units (BPDUs)

The most relevant information in a BPDU is the root path cost

- Every bridge adds its own path cost to the root bridge when receiving a BPDU
- The BPDU is then forwarded to other bridges along the path with the root bridge ID and timers intact
- The only thing that is updated by intermediate switches is the sender bridge ID




In the diagram above, switch A has the lowest mac-address (0000.0000.000a) and will become the root bridge

- After convergence only A will send BPDUs to B and C
- B and C will forward these BPDUs, in this case only on the link between B and C
  - Between B and C another election happens to decide which bridge will put its port in the blocking state (see STP Path Selection below)
  - Both bridges forward a BPDU from A and both have a root path cost of 4
  - The decision is then made based on the lowest bridge ID, which B will win because it has the lowest mac-address (0000.0000.000b)
  - Bridge C will put its port in the blocking state and B will put its port in the forwarding state

## BDPU Types & Info

| Type | Description |
|---|---|
| Configuration | Used for spanning-tree computation |
| Topology Change Notification (TCN) | Used to announce changes in the network topology |

Configuration BPDUs

- Configuration BPDUs are sent out only from Designated Ports
- Designated Ports store the BPDU they send
- Root and Blocking ports store the best BPDU they receive
- Received superior stored BPDUs will expire in MaxAge-MessageAge seconds if not received within this time period

- Access ports send only BPDUs relevant to their access VLAN.
- Trunk ports always send a set of BPDUs E-formatted BPDUs for VLAN1, always untagged.
- PVST+ BPDUs tags for VLAN1, but not for native VLAN
- Original 802.1D (STP) frames are sent in the native VLAN.
    - This is important when questions are asked to send 802.1D frames in VLAN 50'
    - This means that the native VLAN should be set to 50

TCN BPDUs

- Sent only out of root port by non-root bridges
- Carries no data, only announces that something has changed (a port running STP has changed its state)
- Forwarded and acknowledged by every bridge in the network until it reaches the root
- The root will then send a configuration BPDU with the TCN flag informing the entire domain of a change
    - The TCN also makes switches set the aging of mac-address in the cam table from (300 seconds default) to the forward delay (15)
    - This process continues for the max-age time + the forward delay (35 seconds by default)
- TCN BPDUs are not forwarded out of portfast ports and not generated if a portfast port goes up/down
    - For this reason it is important to configure client PC ports as portfast to limit the sending of unnecessary TCN flooding/flushing CAM tables

- RSTP uses the TCN BPDU for backwards compatibility with STP only
    - Topology changes in RSTP are sent via the configuration BDPUs with the TCN flag set

Non-root

```
show spanning-tree detail
 Port 1 (GigabitEthernet1/0/1) of VLAN0001 is root forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.1.
   Designated root has priority 32769, address aabb.cc00.0100
   Designated bridge has priority 32769, address aabb.cc00.0100
   Designated port id is 128.1, designated path cost 0
   Timers: message age 1, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point
   BPDU: sent 3, received 310
```

The 3 BPDUs sent in the output above were sent before STP converged

- After convergence only designated ports should increment their sent counter
- All other ports (root / blocking) should only increment their receive counter
- Likewise the root bridge should not be incrementing its receive counter

## STP Path Selection

1. Lowest root bridge ID
2. Lowest path cost to root bridge
3. Lowest bridge ID
4. Lowest port ID (on the root bridge)

| **Path Attribute** | **Description** |
|---|---|
| Root Bridge ID (RBID) | Sent by root every 2 seconds<br>Root Bridge ID (RBID) is the ID of the root<br>Sender Bridge ID (SBID) is the ID of the root<br>Root Path Cost (RPC) is 0 and<br>Sender Port ID (SPID) is the egress port |
| Root Path Cost (RPC) | Path cost to Root Bridge<br>Set to 0 on the root<br>Other bridges add their own cost to the BPDU when receiving the BPDU (4 in the diagram above) |
| Sender Bridge ID (SBID) | Bridge that originated the BPDU<br>When bridges forward BPDUs from the root, they will update the SBID to their own ID |
| Sender Port ID (SPID) | Egress interface on bridge that originated/forwarded the BPDU<br>When bridges forward BPDUs from the root, they will update the SPID to their own egress port |
| Receiver Port ID (RPID) | Not included in the BPDU message, evaluated locally<br>Port where BPDU was received, not forwarded |

## STP Root Election

Based on the lowest bridge identifier (bridge ID)

- This is a combination of the priority (32768 by default + VLAN-id) and the mac-address
  - The mac-address used is the base mac-address, which is typically 1 number lower than the lowest interface address
  - Find the base mac-address with **show version | i Base**
- Lower numbers are better, a priority of 0 is considered best
  - This can still be overridden by another bridge claiming to be the root if the priority is equal (0) but the base mac-address is lower

## STP Cost Notation

| Bandwidth | Short (old) cost | Long (new) cost |
|---|---|---|
| 4 Mbit/s | 250 | 5000000 |
| 10 Mbit/s | 100 | 2000000 |
| 16 Mbit/s | 62 | 1250000 |
| 100 Mbit/s | 19 | 200000 |
| 1 Gbit/s | 4 | 20000 |
| 2 Gbit/s | 3 | 10000 |
| 10 Gbit/s | 2 | 2000 |

| **spanning-tree pathcost method** *long* \| *short* |
|---|

## STP Port States

| State | Can | Cannot | Duration | Purpose |
|---|---|---|---|---|
| Disabled / Shutdown | - | Send or receive data<br>Send or receive BPDUs | - | Port is administratively shut down |
| Blocking | Receive BPDUs | Send BPDUs<br>Send or receive data | - | Prevents bridging loops and only listens to BPDUs to detect other bridges<br>If bridge loses election on the |

| | | | | |
|---|---|---|---|---|
| | | | | segment, the port will remain in this state |
| Listening | Send and receive BPDUs | Send or receive data<br>Send or receive data | 15 | Bridge thinks the port can either be a root or designated port<br>Port is 'on its way' to begin forwarding<br>Listens for and sends BPDUs and participates in STP |
| Learning | Send and receive BPDUs<br>Receive mac-addresses | Send or receive data | 15 | Same as listening, however bridge will also learn mac-addresses |
| Forwarding | Send and receive BPDUs<br>Send and receive data | - | - | Port sends and receives data<br>Port is either a designated or root port<br>Port sends (or forwards) and receives BPDUs |

- The STP roles can either be root, blocking or designated
  - Root and designated are in the forwarding state
  - Alternate is in the blocking state, alternate is really a RSTP port state, but in IOS it is shown for PVST as well

## STP Link Types

- If a port negotiates half-duplex operation, the switch assumes that the neighbor is a hub (shared link)
- If a port negotiates full-duplex operation, the switch will assume that the neighbor is a switch (P2P link)

| |
|---|
| **spanning-tree link-type** *shared* \| *point-to-point* |

These link types do not really matter for PVST+/CST, and primarily come into play when running RSTP/RPVST+

## STP Timers

| Timer | Duration | Description |
|---|---|---|
| MessageAge | - | Estimation of BPDU age since it was originated by root bridge (0 at root) |

| | | |
|---|---|---|
| | | Every bridge increments this by 1 before forwarding the BPDU |
| MaxAge-MessageAge | - | Remaining lifetime of a BPDU after being received by a bridge<br>If port stores BPDU it must be received again within the MaxAge-MessageAge interval |
| Hello | 2 | Root switch creates and sends a hello every 2 seconds (default)<br>Hellos are always received on root port<br>Forwarded out designated ports with RPC, SBID, SPID, and MessageAge fields updated |
| Max-Age | 20 | Maximum time a BPDU is stored<br>The time a bridge waits after it stops receiving BPDUs (from the root) on the specified port |
| Forward-Delay | 15 | Time spent in listening and learning state |

- All timers are set by the root bridge and should only be updated on the root bridge
  - You can configure different timers on non-root bridges, but they will only be used when they become the root
  - Non-root bridges will use the timers set by the root, regardless of their locally configured values
- The forward delay is set to 15 seconds by default and applies to both the learning and listening state (15 sec for learning, 15 sec for listening)
  - You cannot configure alternating timers for the learning and listening states

**spanning-tree vlan** *vlan-id* **hello-time** *seconds*
**spanning-tree vlan** *vlan-id* **forward-time** *seconds*
**spanning-tree vlan** *vlan-id* **max-age** *seconds*

```
spanning-tree hello-time 1
spanning-tree hello-time 4
spanning-tree max-age 10
```

- If you don't specify a VLAN ID the timers will apply to all VLANs

# BackboneFast / UplinkFast

## UplinkFast (802.1D STP)

- Deals with direct failures, basically RSTP alternate port on IEEE STP
    - The switch in the access layer usually have two uplinks, one root port and one blocking
    - The switch knows the domain is loop-free by the BPDUs received on the blocking port
    - Uplinkfast is triggered when the root port goes down or the BPDUs stop
    - The switch will move over to the blocking port and change it to the new root port
- Blocked ports immediately transition to the forwarding state by skipping listening and learning state
- When you enable UplinkFast the priority for all VLANs is set to 49152
- The spanning tree port cost is increased by 3000 (short) or 10^7 (long)
- UplinkFast is enabled / disabled for all VLANs
    - Only actually active if switch has a STP blocking port
    - Use on access layer switches
    - Never try to enable on root, no error messages are generated when you do

spanning-tree uplinkfast

- After uplinkfast is triggered, the bridge will also start flooding its cam table to its neighbors (using ARP)
    - The switch does this by spoofing the known mac-addresses to the L2 multicast destination 0100.0ccd.cdcd
    - The source of these frames will be spoofed by the switch and will be exactly the same mac-address as the hosts connected to the switch
    - 1 spoofed frame will be sent per entry in the CAM table
- The point of this flooding is a result of using a new root port to a new uplink bridge
    - The new uplink bridge does not know the contents of the current mac-address table because the port was previously blocking
    - By spoofing the CAM table, the new uplink bridge learns the contents of the CAM table which will prevent unnecessary unicast flooding
    - The rate at which this happens is configurable with the **spanning-tree uplinkfast max-update-rate** command (default is 150 packets/sec)

**spanning-tree uplinkfast max-update-rate** *packets/sec*

```
show spanning-tree uplinkfast
UplinkFast is enabled

Station update rate set to 300 packets/sec.

UplinkFast statistics
-----------------------
Number of transitions via uplinkFast (all VLANs)            : 0
Number of proxy multicast addresses transmitted (all VLANs) : 0

Name                 Interface List
-------------------- ------------------------------------
VLAN0001             Gi1/0/1(fwd), Gi1/0/2
```

```
show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     aabb.cc00.0100
              Cost        3004
              Port        1 (GigabitEthernet1/0/1)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    49153  (priority 49152 sys-id-ext 1)
              Address     aabb.cc00.0300
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec
  Uplinkfast enabled

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------
Gi1/0/1             Root FWD 3004      128.2    P2p
Gi1/0/2             Altn BLK 3004      128.1    P2p
```

## BackboneFast (802.1D STP)

- Deals with indirect failures
- Max age is the timer a bridge waits after it stops receiving BPDUs (from the root) on the specified port
    - Max age unblocks links after 20 seconds, or when 10 BPDUs are missed, in the meantime no BPDUs are sent out of this port
    - Under normal condition this failure process takes 20 seconds + listening + learning = 50 seconds
- These BPDUs are sent by the isolated switch that declared itself root

    - During the max age (20 seconds) the falsely declared root will not receive any BPDUs from its neighbor because the port is blocking
    - The switch that receives these BPDUs correctly flags these 'root' BPDUs as inferior and will keep the port blocked for the max-age

```
spanning-tree backbonefast
```

BackboneFast removes the max-age timer if inferior BPDUs are received on a (separate) blocked port

- The switch will first generate a RLQ (Root Link Query) to the root bridge to verify that the other switch has lost connectivity to the root
    - The other switch is isolated and thus declares itself root
    - The root will respond to the RLQ and the switch will remove max-age timer from the blocked port
    - The port still has to go through the listening and learning state (30 seconds)
- Must be enabled on all switches in the network
    - Not supported on Token Ring networks
    - Only switches that have blocked ports will generate a RLQ

# MSTP

## Multiple Spanning Tree Protocol (802.1S MSTP)

- You can configure multiple MST regions within the same topology
    - The region is defined by the name
    - The revision number must match between neighbors
    - Instance to VLAN mappings are not sent of the link, instead a MD5 hash is sent and compared between bridges
    - Rely on either VTP or manual configuration to configure the same instance to VLAN mappings between bridges
    - You can configure instances in the range of 1-4094, however only 16 (MST0 + 15 user defined) total are supported on cisco switches

- MSTP uses the long path cost notation by default
- Common Spanning Tree (CST) is the entire spanning-tree domain including other versions of STP
    - Inside the MST region the IST (Internal STP) is present, this is MST0
    - The IST is the only STP instance that sends and receives BPDUs

- Hello, ForwardTime and MaxAge timers can only be tuned for the IST
    - All other MIST inherit the timers from the IST
    - The hello-time is the timer at which configuration messages are sent by the root switch

- Configure the maximum number of switches between any two end stations with the diameter command (MST0 only)
- The MST extended system-id is made up of the instance number instead of the VLAN number (instance 1 with default priority is 32679)

```
spanning-tree mode mst
spanning-tree mst configuration
 name cisco
 revision 1
 instance 1 vlan 100-200
 show pending

spanning-tree mst 0 root primary diameter 7 hello-time 2
spanning-tree mst 1 priority 28672
```

```
show pending
Pending MST configuration
Name      [cisco]
Revision  1     Instances configured 2

Instance  Vlans mapped
--------  ----------------------------
0         1-99,201-4094
1         100-200
---------------------------------------
```

```
show spanning-tree mst configuration
Name      [cisco]
Revision  1     Instances configured 2

Instance  Vlans mapped
--------  ----------------------------
0         1-99,201-4094
1         100-200
---------------------------------------
```

```
show spanning-tree mst
##### MST0    vlans mapped:   1-99,201-4094
Bridge        address aabb.cc00.0100  priority      24576 (24576 sysid 0)
Root          this switch for the CIST
Operational   hello time 2 , forward delay 15, max age 20, txholdcount 6
Configured    hello time 2 , forward delay 15, max age 20, max hops    20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/0/1          Desg FWD 4         128.3    P2p Bound(PVST)

##### MST1    vlans mapped:   100-200
Bridge        address aabb.cc00.0100  priority      28673 (28672 sysid 1)
Root          this switch for MST1

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/0/1          Desg FWD 4         128.3    P2p Bound(PVST)
```

MST Configuration with VTPv3

```
vtp version 3
vtp mode server mst
vtp primary mst force
```

## MST Caveats

If one VLAN is active on the link in the instance, the entire instance is active

- This can be a problem if other VLANs are pruned (administratively disallowed) that are part of the instance

Make sure that all VLANs that are part of the instance are allowed on the trunk

- If not, either add the VLAN to the allowed list (if not restricted)
- Or remove other VLANs from the allowed list (prune them) so the MST instance switches to another link
- Or put the disallowed VLAN in another instance by itself and make it go over a separate link

## MST Interoperability

The idea is to hide the internal MST from other versions of STP. But maintain backwards compatibility and fast convergence

- Other regions will view the MST region as a single switch
- Every MST region runs a special instance of spanning-tree known as IST or Internal Spanning Tree (MST0)
- IST has a root bridge, elected based on the lowest Bridge ID

With multiple MST regions in the network, a switch that receives BPDUs from another region is a boundary switch

- Another region can be RSTP or PVST+, the ports to these regions are marked as MST boundary ports
    - Shown as P2p Bound(STP) in **show spanning-tree**
- When multiple regions connect together, every region constructs its own IST and all regions build a common CIST
- A CIST Root is elected among all regions and CIST Regional Root (IST root) is elected in every region

| **Root Type** | **Description** |
|---|---|
| CIST Root | The bridge that has the lowest Bridge ID among ALL regions<br>This could be a bridge inside a region or a boundary switch in a region |
| CIST Regional Root | A boundary switch elected for every region based on the shortest external path cost to reach the CIST Root<br>Path cost is calculated based on costs of the links connecting the regions, excluding the internal regional paths<br>CIST Regional Root becomes the root of the IST for the given region as well |

# Loopguard / UDLD

## Protecting Against Sudden Loss of BPDUs

Loopguard and UDLD protect against a sudden loss of BPDUs from a neighboring bridge

- This can be dangerous because switches might (falsely) think that a topology changed has happened
- This behavior can lead to bridging loops when there is no actual TCN and the BPDUs are just delayed / timed out
- UDLD is a better option for etherchannels, because it can block individual ports

Protects from

- Unidirectional links
- Switches/links that block BPDUs

## STP Loopguard

- Prevents alternate or root ports from becoming designated in response to a unidirectional link
- Puts ports into a loop-inconsistent state if BPDUs are no longer received on the interface (basically blocking state)

    - Does not block the entire port from receiving data, only the VLANs on which STP is active
- Incompatible with RootGuard
- Prevents loops by detecting the sudden loss of BPDUs
    - Only superior BPDUs are considered
    - Loopguard-enabled ports may send BPDUs when inferior BPDUs are received

```
spanning-tree loopguard default

int gi1/0/1
 spanning-tree guard loop

show spanning-tree inconsistentports
```

# Portfast + Features

## STP PortFast (Edge Port)

Interface is moved directly to forwarding state, bypassing learning/listening without forward-time delay (15sec x2)

- This only happens when the port transitions from a disabled state, not from a blocking / alternate state
- PortFast ports will still transmit BPDUs, but will lose their PortFast state if a BPDU is received
- TCNs are not generated when a portfast port changes state (important)
    - This means that connecting/disconnecting hosts on portfast ports will not lead to STP topology re-calculations
- Shown as P2P edge in the show commands

When configured globally PortFast will be enabled on all access ports

- Just because PortFast is configured on the port or in the global configuration, doesn't mean that it is operational
- Cisco states that enabling portfast on links connecting to other switches will lead to temporary bridging loops
    - This is technically true, but portfast ports will lose their state if a BPDU is received (a very brief moment between switches)
- When the interface comes online, portfast will skip the listening and learning state and will become designated forwarding
    - During this time (before the first BPDU is received) a small bridging loop may exist
    - This bridging loop will disappear when the first BPDU is received and is in my opinion negligible
    - Remember that this state will only occur when the state is changed from disabled, not blocking / alternate (no shutdown)

- Also remember that this will only happen by default on access ports, and inter-switch links are almost always trunks
- If asked on exam, yes configuring portfast between switches will always unconditionally lead to bridging loops that kill your network

```
spanning-tree portfast default

interface gi1/0/1
 spanning-tree portfast
 spanning-tree portfast disable
 spanning-tree portfast trunk
```

- **disable** - Use this in combination with **spanning-tree portfast default** to exclude certain access ports from the default state
- **trunk** - Enable PortFast on trunk ports (will also turn on portfast on access ports)
  - PortFast on trunks works on a per-VLAN basis
  - Can be off for certain VLANs but on for others (depending on root placement / reception BPDUs for specific VLANs)

```
show spanning-tree interface gi1/0/1 detail
 Port 3 (GigabitEthernet1/0/1) of VLAN0001 is designated forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.1.
   Designated root has priority 24577, address aabb.cc00.0100
   Designated bridge has priority 24577, address aabb.cc00.0100
   Designated port id is 128.1, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   The port is in the portfast mode
   Link type is point-to-point
   BPDU: sent 3, received 0
```

```
show spanning-tree interface gi1/0/1
Vlan                Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ---------
VLAN0001            Desg FWD 4         128.1    P2p Edge
```

## STP BPDUGuard

- BPDUGuard supersedes PortFast and can be configured per port or globally
  - Errdisables port if BPDU is received, can be automatically enabled with **errdisable recovery cause bpduguard**
- Global BPDUGuard is part of PortFast and will only be active if PortFast is also enabled (operational)

    - It is very important to understand that this is the operational state of PortFast, not the administrative state (see PortFast above)
- When configured on the port, BPDUGuard is enabled unconditionally and does not need PortFast to function

```
spanning-tree portfast bpduguard default

int gi1/0/1
 spanning-tree bpduguard enable

errdisable recovery cause bpduguard
errdisable recovery interval 30
```

```
*Dec 30 08:33:37.480: %SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port Gi1/0
*Dec 30 08:33:37.480: %PM-4-ERR_DISABLE: bpduguard error detected on Gi1/0/1,

show interfaces status err-disabled
Port      Name               Status       Reason               Err-disabled Vl
Gi1/0/1                      err-disabled bpduguard
```

## STP BPDUFilter

- Filters BPDUs on the port in egress direction (global) or both directions (per interface)

Global BPDUFilter stops sending outgoing BPDUs on interfaces that have an operational PortFast status

- Will still sent 11 BPDUs when the interface comes online, this is to prevent misconfigurations
- PortFast enabled ports that receive BPDUs will lose their PortFast status, and thus global BPDUFilter will also be disabled

BPDUFilter configured on the port itself will filter all incoming and outgoing BPDUs

- This is the equivalent of turning off STP on the port
- PortFast does not have to be enabled on the port for BPDUFilter to be active

```
spanning-tree portfast bpdufilter default

int gi1/0/1
 spanning-tree bpdufilter enable
```

## STP Interoperability PortFast, BPDUFilter and BPDUGuard

Global BPDUFilter + BPDUGuard

- Guard is triggered first
- PortFast triggered second
- Filter is triggered third

This is a valid configuration that err-disables ports that receive BPDUs, and filters outgoing BPDUs

Per port BPDUFilter + BPDUGuard:

- Filter is applied first
- Guard is triggered second
- PortFast is applied third

BPDUFilter configured on the port itself will filter BPDUs in both directions, and will supersede PortFast and BPDUGuard on the port

- This is not a valid configuration because incoming and outgoing BPDUs will be filtered by BPDUFilter
- The guard never receives BPDUs and will never be triggered

# Root Guard

## STP Rootguard

Ignores superior BPDUs on specified ports, preventing rogue switches from becoming the root

- Apply on interfaces that connect to switches that should never become the root
- Allows interface to participate in STP but will trigger when superior BPDUs are received
    - Rootguard ignores superior BPDUs and will put a port in a 'root-inconsistent' state (except MST)
    - This state does not allow sending or receiving of data, only BPDUs (basically the listening state)
    - This state is automatically cleared when superior BPDUs stop

- In MST the port does not become root-inconsistent, but is forced to become a designated port
- Not VLAN-aware, enabled for all VLANs present on the port
    - Cannot be enabled globally, disabled by default
- Do not enable the root guard on interfaces to be used by the UplinkFast feature
    - When combined with UplinkFast the blocking ports will go to root-inconsistent state instead of forwarding state

```
int gi1/0/1
 spanning tree guard root
```

```
show spanning-tree interface gi1/0/1 detail
 Port 1 (GigabitEthernet1/0/1) of VLAN0001 is designated forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.1.
   Designated root has priority 32769, address aabb.cc00.0100
   Designated bridge has priority 32769, address aabb.cc00.0100
   Designated port id is 128.1, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point
   Root guard is enabled on the port
   BPDU: sent 672, received 8
```

# RSTP (RPVST+)

## Rapid Per-VLAN Spanning Tree Plus RPVST+ (Cisco / 802.1W)

- The + means that the STP instance is backwards compatible with IEEE standard STP
  - Per-VLAN instances appear as one instance to non-cisco switches

RSTP uses BPDU version 2 instead of version 0 in the Message Type field, to distinguish from 802.1D STP

- Will start speaking 802.1D STP on links if detected for backwards capability
  - These links will show as P2P Peer (STP) in **show spanning-tree**
- BPDUs are sent out every switch port at hello time intervals,
  - This happens regardless of whether BPDUs are received from the root
  - If three BPDUs are missed in a row, the neighbor is presumed down and is aged out

## STP Port States

| State | Can | Cannot | Duration | Purpose |
|---|---|---|---|---|
| Discarding | Receive BPDUs | Send or receive data<br>Send BPDUs | - | Basically combines STP disabled, blocking and listening state into one<br>Listens for BPDUs |
| Learning | Send and receive BPDUs<br>Receive mac-addresses | Send or receive data | 15 | Same as listening, however bridge will also learn mac-addresses |
| Forwarding | Send and receive BPDUs<br>Send and receive data | - | - | Port sends and receives data<br>Port is either a designated or root port |

| | | | | |
|---|---|---|---|---|
| | | | | Port sends (or forwards) and receives BPDUs |

- The RSTP roles can either be root, backup, alternate or designated
  - Root and designated are in the forwarding state
  - Alternate is in the blocking state, but is a prospective replacement for the root port (basically uplinkfast)
  - Backup is in the blocking state, but is a prospective replacement for the designated port on a shared segment

### RSTP Port Types

| Port Type | Function |
|---|---|
| Edge | PortFast port to a single host |
| Root | Best cost to the root bridge, only 1 per VLAN |
| P2P | Any port that connects to another bridge |

- Point-to-point ports are defined based on their duplex value (full-duplex)

### RSTP Synchronization

- RSTP is not faster because of increased timers, but because of synchronization
  - The synchronization process only works on ports that are P2P and full-duplex
  - RSTP uses these ports to form a 'handshake' with the neighboring bridge, knowing that there is only 1 device at the other end
    RSTP detects a topology change only when a non-edge port transitions to the Forwarding state, not when STP enabled ports go up or down

# STP Root Configuration

### Bridge-ID

- Traditional 802.1D bridge priority value (16 bits), followed by the MAC address for the VLAN
- The 802.1t extended system ID (4-bit priority multiplier, plus a 12-bit VLAN ID), followed by the MAC address for the VLAN

| **spanning-tree extend system-id** |
|---|

If the switch cannot support 1024 unique MAC addresses for its own use, the extended system ID is always enabled by default

- Otherwise, the traditional method is enabled by default

The actual priority of a default STP bridge for VLAN1 is 32768+1 = 32769 and so on…

## Root Allocation

**spanning-tree vlan** *vlan-list* **priority** *bridge-priority*

Priority has to be set in increments of 4096

**spanning-tree vlan** *vlan-id* **root** *[ primary | secondary ]* **diameter** *diameter* **hello-time** *seconds*

**Root primary**

- If the current root priority is more than 24576, the local switch sets its priority to 24576 (32768 - 2 x 4096)
- If the current root priority is less than 24576, the local switch sets its priority to 4096 less than the current root
    - Does not work if the current root has a priority of 4096, use the **priority 0** command instead

**Root secondary**

- Does not detect other potential root bridges and will blindly set the priority to 28672 (32768 - 1 x 4096)

**Diameter**

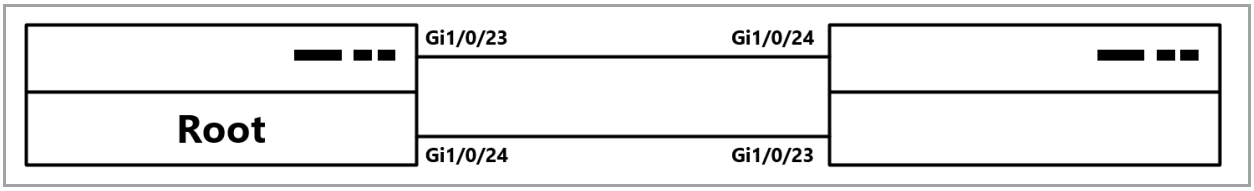
- You can view this as just a simple hop count of how large the switching domain is
- Default value is 7, range is 1-7
- All this applies to is the automatic configuration of the hello-timer based on the network diameter

## STP Cost & Priority

- Configure priority settings to influence the remote device (lower is better)
    - The port-id is actually a combination of the interface number (mac-address) and the priority
    - Configure on root bridges or uplink switches to the root bridge
- Configure cost settings to influence the local device (lower is better)
    - Cost is used over priority settings
    - Configure on non-root bridges, not the root because the root has no cost to itself

**spanning-tree port-priority** *priority*
**spanning-tree vlan** *vlan-id* **cost** *cost*

- The cost command does not care whether the path-cost method is long or short
- If you don't specify a VLAN ID the cost will apply to all VLANs




By default the lower port-ID (gi1/0/23) on the root will be used as the active path

- Influence this decision by lowering the priority on gi1/0/24 on the root, or increasing the priority on gi1/0/23 on the root
- Influence this decision by lowering the cost on gi1/0/23 on the non-root, or increasing the cost on gi1/0/24 on the non-root

```
show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     aabb.cc00.0100
              Cost        4
              Port        4 (GigabitEthernet1/0/24)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     aabb.cc00.0200
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------
Gi1/0/23            Altn BLK 4         128.23   P2p
Gi1/0/24            Root FWD 4         128.24   P2p
```

Root

```
interface gi1/0/24
 spanning-tree port-priority 64
```

```
show spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     aabb.cc00.0100
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     aabb.cc00.0100
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- ----------------------
Gi1/0/23            Desg FWD 4         128.23   P2p
Gi1/0/24            Desg FWD 4          64.24   P2p
```

Non-root

```
interface gi1/0/24
 spanning-tree cost 2
```

```
show spanning-tree interface gi1/0/24 detail
 Port 4 (GigabitEthernet1/0/24) of VLAN0001 is root forwarding
   Port path cost 2, Port priority 128, Port Identifier 128.24.
   Designated root has priority 32769, address aabb.cc00.0100
   Designated bridge has priority 32769, address aabb.cc00.0100
   Designated port id is 128.23, designated path cost 0
   Timers: message age 16, forward delay 0, hold 0
   Number of transitions to forwarding state: 2
   Link type is point-to-point
   BPDU: sent 8, received 62
```

- The .23 under the priority number is the port id (priority + interface number)

# STP Types

## Spanning-Tree Protocol (STP) Types

| STP Type | Abbreviated | Standard | Description |
|---|---|---|---|
| IEEE Spanning Tree Protocol | STP | 802.1D | General standard |

| | | | |
|---|---|---|---|
| Common Spanning Tree | CST | 802.1Q | A single instance of STP that encompasses all VLANs |
| Per-VLAN Spanning Tree | PVST | Cisco | Separate instance of STP for each individual VLAN<br>Requires ISL, not compatible with other STP types |
| Per-VLAN Spanning Tree Plus | PVST+ | Cisco | Separate instance of STP for each individual VLAN<br>Compatible with other STP types, including PVST and CST |
| Rapid Per-VLAN Spanning Tree | RSTP | 802.1W | Enhanced STP using synchronization<br>A single instance of STP that encompasses all VLANs<br>Backwards compatible with other STP types |
| Rapid Per-VLAN Spanning Tree Plus | RPVST+ | Cisco | Enhanced STP using synchronization<br>Separate instance of STP for each individual VLAN<br>Backwards compatible with other STP types |
| Multiple Spanning Tree | MST | 802.1S | Groups multiple VLANs into STP regions<br>Backwards compatible with other STP types (boundary) |

# Switching Features

# Bridge Groups

## Bridge Groups (Fallback Bridging)

- Allows non-IP traffic to be communicated between hosts that reside in different VLANs or are connected on routed ports
- SVIs allow hosts in different VLANs to communicate using IP addressing. Bridge groups extends this functionality to L2 as well
- Can only be applied to SVI interfaces or L3 routed ports (no switchport)
- An actual IP address is not required to be configured on these SVIs or routed ports for fallback bridging to work

```
bridge 1 protocol vlan-bridge
int fa0/0
 no switchport
 bridge group 1

int vlan 10
 bridge-group 1

show bridge 1 group
```

# CDP

## Cisco Discovery Protocol (CDP)

- CDP messages (advertisements) are sent to the Cisco proprietary multicast address 0100.0ccc.cccc
    - Messages are sent every 60 seconds by default
    - The hold-timer is 3x the advertisement timer (180 seconds by default)
- CDP carries various pieces of information in Type-Length-Value fields (TLVs)
- CDP is enabled by default and will most likely be CDPv2 (see below)
- CDP can be used for routing (see On Demand Routing / ODR section)

Globally enable CDP

```
cdp run
```

- Disable cdp globally on all ports with **no cdp run**

Enable CDP on interface

```
interface
 cdp enable
```

- Disable cdp on specific port with **no cdp enable**

Customize CDP timers

```
cdp timer seconds
cdp holdtime seconds
```

```
cdp timer 20
cdp holdtime 60
```

```
show cdp interface gi1/0/1
GigabitEthernet1/0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 20 seconds
  Holdtime is 60 seconds
```

Clear CDP information

```
clear cdp counters
clear cdp table
```

## CDP Type-Length-Values (TLVs)

| TLV | CDPv1 / CDPv2 | Definition |
|---|---|---|
| Device-ID | 1 2 | The device name |
| Address | 1 2 | IP addresses configured on the link (including secondary)<br>In case of switch it will be an address of an SVI |
| Port-ID | 1 2 | Source port |
| Prefix | 1 2 | Subnets present on the device (including SVIs) |
| Capabilities | 1 2 | Device features |
| Version | 1 2 | IOS version |
| Platform | 1 2 | Hardware |
| VTP domain | 2 | VTP domain name |
| Native VLAN | 2 | VLAN-id of the native VLAN on dot1q trunks |
| Duplex | 2 | Duplex mode (half or full) |

- CDP can also transfer information regarding power discovery and PoE classes

You can customize which information gets sent to neighbors by creating custom TLV lists

- Applies globally for all neighbors or per interface
- Can only apply a single custom TLV list globally per device
- VTP domain and native VLAN always seem to be included, regardless of limitation (unverified)

```
cdp tlv-list list_name
 tlv 1....
 tlv 2....
```

```
cdp filter-tlv-list list_name

interface
 cdp filter-tlv-list list_name
```

**tlv-list** - Specify the information that gets transferred to a neighbor
**filter-tlv-list** - Apply the list globally to the device or to an interface

Globally apply and transfer mgmt-address and duplex info

```
cdp tlv-list LIMIT
 mgmt-address
 duplex

cdp filter-tlv-list LIMIT
```

```
show cdp tlv-list LIMIT
Tlv-list : LIMIT
 Version
Applied on: Globally
```

Only send version and apply on interface

```
cdp tlv-list LIMIT
 version
```

```
interface gi1/0/1
 cdp filter-tlv-list LIMIT
```

```
show cdp interface gi1/0/1
GigabitEthernet1/0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
  Tlv-list applied : LIMIT
```

## CDPv1 vs CDPv2

The version generally in use is CDPv2 which includes reporting that allows for more rapid error tracking

- CDPv2 also supports the following TLVs
  - Duplex information
  - Native VLAN information
  - VTP domain information
- CDPv2 Errors messages can be sent to the console or to a logging server, includes:
  - Duplex errors
  - Native VLAN mismatches

Go back to CDPv1

```
no cdp advertise-
v2
```

- This stops CDP from reporting any errors on the link relating to duplex/native VLAN errors
  - Even if it is disabled on one of the two neighbors, the error reporting will stop

```
show cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is not enabled
```

CDPv2 packet

```
Cisco Discovery Protocol
    Version: 2
    TTL: 180 seconds
  > Checksum: 0x741d [correct]
  > Device ID: CSW1
  > Software Version
  > Platform: Cisco
  > Addresses
  > Port ID: GigabitEthernet0/1
  v Capabilities
      Type: Capabilities (0x0004)
      Length: 8
    v Capabilities: 0x00000029
        .... .... .... .... .... .... .... ...1 = Router: Yes
        .... .... .... .... .... .... .... ..0. = Transparent Bridge: No
        .... .... .... .... .... .... .... .0.. = Source Route Bridge: No
        .... .... .... .... .... .... .... 1... = Switch: Yes
        .... .... .... .... .... .... ...0 .... = Host: No
        .... .... .... .... .... .... ..1. .... = IGMP capable: Yes
        .... .... .... .... .... .... .0.. .... = Repeater: No
  > VTP Management Domain: cisco
  > Native VLAN: 1
  > Trust Bitmap: 0x00
  > Untrusted port CoS: 0x00
  > Management Addresses
```

CDPv1 packet

```
Cisco Discovery Protocol
    Version: 1
    TTL: 180 seconds
  > Checksum: 0x24d7 [correct]
  > Device ID: CSW1
  > Software Version
  > Platform: Cisco
  > Addresses
  > Port ID: GigabitEthernet0/1
  v Capabilities
      Type: Capabilities (0x0004)
      Length: 8
    v Capabilities: 0x00000029
        .... .... .... .... .... .... .... ...1 = Router: Yes
        .... .... .... .... .... .... .... ..0. = Transparent Bridge: No
        .... .... .... .... .... .... .... .0.. = Source Route Bridge: No
        .... .... .... .... .... .... .... 1... = Switch: Yes
        .... .... .... .... .... .... ...0 .... = Host: No
        .... .... .... .... .... .... ..1. .... = IGMP capable: Yes
        .... .... .... .... .... .... .0.. .... = Repeater: No
  > IP Prefixes: 1
```

## CDP Output

```
show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
CSW1             Gig 1/0/1         179             R S I            Gig 1/0/1
```

- The local interface is the interface on the device
- The port ID is the interface on the neighbor device

The capabilities reflect those in the CDP packet capture and the output below:

| **Capability** | **Short name** |
|---|---|
| Router | R |
| Switch | S |
| Transparent Bridge | T |
| Source Route Bridge | B |
| Host | H |
| IGMP capable | I |
| Repeater | r |
| CVTA | C |
| Phone | P |
| Remote | D |
| Two-port Mac Relay | M |

```
show cdp neighbors detail

-------------------------
Device ID: CSW2
Entry address(es):
  IP address: 10.0.0.2
Platform: Cisco ,  Capabilities: Router Switch IGMP
Interface: GigabitEthernet0/0,  Port ID (outgoing port): GigabitEthernet0/0
Holdtime : 53 sec

Version :
Cisco IOS Software

advertisement version: 2
VTP Management Domain: 'cisco'
Native VLAN: 1
Duplex: half
Management address(es):
  IP address: 10.0.0.2
```

- All information listed here is the neighbor information, including the interface and outgoing port ID

# Etherchannel (LAG)

## Etherchannels (Link Aggregation Groups / LAG)

- Bundle up to eight (active) links of the same type into a logical Link Aggregation Group (LAG)
  - 8x Fast Etherchannel (FEC) = throughput of up to 1600 Mbps (2 directions)
  - 8x Gigabit Etherchannel (GEC) = throughput of up to 16 Gbps (2 directions)
  - 8x 10-Gigabit Etherchannel (10GEC) = throughput of up to 160 Gbps (2 directions)
- A multi-chassis Etherchannel (MEC) is a LAG divided between multiple members of a stack
- Even though the (potential) throughput is 8 times higher, doesn't mean the speed is 8 times higher as well
  - Load is shared between the links and this is not always done equally (depending on algorithm)

- Only one interface will be 'master in the bundle', contrary what **show etherchannel detail** will show
  - This interface is responsible for reception and transmission of STP frames
  - The interface in the bundle with lowest port-id is usually the master in the bundle

## Load-Balancing Algorithms

- Traffic is not sent over multiple paths (ECMP) instead a link in the bundle is chosen
  - This is basically load-sharing and is entirely dependent on the algorithm

    - The algorithm is just a hash value based on the load-balancing method (default is usually src-mac)
    - Different traffic-types or network designs can lead to different algorithms
    - Think about traffic paths (including return) when choosing the optimal algorithm
    - Even if you chose the optimal pattern, hosts may use different amounts of traffic, resulting in an imbalance
- Simple algorithms use only a single piece of information to make a choice
    - The same traffic will always use the same link
- Complex algorithms use multiple pieces of information to make a choice
    - The same traffic will always use the same link
- Not all switches may support all load-balancing methods, largely dependent on platform
    - When using ip-based hashing the switch will fall back to the next lowest method when non-ip traffic is received
- Neighboring switches do not have to match on load-balancing methods

| port-channel **load-balance** *method* |
|---|

Use **show etherchannel port-channel** to acquire an insight into the link utilization

## Load-balancing Methods

| **Method** | **Simple/Complex** | **Choice made based on** | **Hash** |
|---|---|---|---|
| SRC-MAC | Simple | Source mac-address only | Bit |
| DST-MAC | Simple | Destination mac-address only | Bit |
| SRC-IP | Simple | Source ip-address only | Bit |
| DST-IP | Simple | Destination ip-address only | Bit |
| SRC-PORT | Simple | Source port only | Bit |
| DST-PORT | Simple | Destination port only | Bit |
| SRC-DST-MAC | Complex | Source and destination mac-address | XOR |
| SRC-DST-IP | Complex | Source and destination ip-address | XOR |
| SRC-DST-PORT | Complex | Source and destination port | XOR |

The complex load-balancing methods use a XOR operation to make a choice on which link to take
Links are chosen based on the last bits in the hash

- 2-link etherchannels use only the last bit (0 or 1)
- 4-link etherchannels use the last two bits (00, 01, 10 or 11)
- 8-link etherchannels use the last three bits (000, 001, 010, etc.)

Simple load-balancing methods make a decision simply on the last bit value (0 or 1)

2-link Etherchannel

| Value 1 | Value 2 | XOR Outcome | Ethernet Link chosen |
|---|---|---|---|
| 0 | 0 | 0 | gi1/0/1 |
| 0 | 1 | 1 | gi1/0/2 |
| 1 | 0 | 1 | gi1/0/2 |
| 1 | 1 | 0 | gi1/0/1 |

Example #1
Load-balancing method is source-destination-ip and two etherchannel links are used (gi1/0/1-2)

- Source ip-address is 10.0.0.1
- Destination ip-address is 172.16.0.254

Source ip-address xxx.xxx.xxx.0000001 = last bit = 1
Destination ip-address is xxx.xxx.xxx.01111111 = last bit = 1

XOR outcome = 1 XOR 1 = 0 = ethernet link gi1/0/1

Example #2
Load-balancing method is source-destination-ip and eight etherchannel links are used (gi1/0/1-8)

- Source ip-address is 10.0.0.1
- Destination ip-address is 172.16.0.254

Source ip-address xxx.xxx.xxx.0000001 = last three bits = 001
Destination ip-address is xxx.xxx.xxx.01111111 = last three bits = 111

XOR outcome = 001 XOR 111 = 110 = ethernet link gi1/0/7

```
port-channel load-balance src-dst-ip
```

```
show etherchannel load-balance
 EtherChannel Load-Balancing Configuration:
        src-dst-ip
 EtherChannel Load-Balancing Addresses Used Per-Protocol:
 Non-IP: Source XOR Destination MAC address
   IPv4: Source XOR Destination IP address
   IPv6: Source XOR Destination IP address
```

## STP Etherchannel Guard

- Detect etherchannel misconfiguration between the switch and a connected device (or another switch)
- Places interfaces in errdisable state in the event of a misconfiguration
  - This applies to etherchannels configured with mode on
  - LACP and PAgP have mechanisms in place to detect misconfigurations, mode on does not
- The detection of a misconfig depends on the duplicate reception of BPDUs on a port-channel interface
  - This is dependent on which bridge is the root of spanning tree
- On by default

```
spanning-tree etherchannel guard misconfig
```

# LACP

## Link Aggregation Control Protocol (LACP)

- 16 interfaces can be bundled into a PAgP etherchannel, but only 8 will be active (see Hot-Standby below)
- LACP does not support half-duplex ports, half duplex ports will be put in suspended state
- LACP frames (hello) are sent at a 30 second interval to the L2 multicast address 0180.c200.0002
- Bundled interfaces need to match on:
  - Speed
  - Duplex
  - VLANs
  - Trunk / Access
- Bundled interfaces do not need to match on
  - STP cost (if hardcoded)
  - Bandwidth (if hardcoded) - Port channel will be torn down at regular intervals because bandwidth does not match
  - LACP mode (active or passive)
- LACP Suspended ports (not meeting above criteria) will transition to STP blocking ports

- One switch will become the LACP master and one will become the slave
  - This decision is based on the system-id (same as STP, mac-address + priority 32768 by default)
  - Usually the winner of STP on the segment (with default config/priority) will also become the LACP master
  - Manually change the priority with the **lacp system-priority** global command (lower priority is better)

- The master decides which ports in the bundle are active and which are hot-standby (when using 8+ ports)

## LACP Hot-Standby

- Additional links (up to 8) are placed in hot-standby mode to replace links that have become inactive
  - 16 interfaces total of which 8 are active, and 8 are hot-standby
  - The decision on which links are active are based on port-id and priority (lower priority is better)
  - The master switch has control over which ports are active (master switch port-id is used)
- The priority is equal (32768) by default, meaning that the highest port-id's will become hot-standby

```
lacp system-priority 0

interface range gi1/0/1 - 8
 lacp port-priority 200

interface range gi1/0/9 - 16
 lacp port-priority 100
```

- Switch will become the master, interfaces gi1/0/9 - 16 will be active and gi1/0/1 - 8 will be hot-standby

## LACP Modes

| Mode | Description |
|---|---|
| Active | Actively tries to negotiate etherchannel by continuously (30sec) sending LACP frames |
| Passive | Passively negotiates etherchannel by listening for LACP frames |

- You can mix and match different modes on interfaces in the same bundle

This is a valid configuration

```
interface range gi1/0/1 - 4
 channel-group 12 mode active

interface range gi1/0/5 - 8
 channel-group 12 mode passive
```

**channel-protocol** lacp
**channel-mode** *1-255* **mode** *active / passive*

- The **channel-protocol** command is not required
  - If it is defined however, you cannot use mode on or PAgP on the link

```
interface range gi1/0/1 - 2
 shutdown
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 12 mode active

interface port-channel 12
 switchport trunk encapsulation dot1q
 switchport mode trunk

interface range gi1/0/1 - 2
 no shutdown
```

```
show lacp sys-id
32768, aabb.cc00.0100
```

```
show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 12
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi1/0/1     SA      bndl      32768         0xC       0xC     0x1         0x3D
Gi1/0/2     SA      bndl      32768         0xC       0xC     0x2         0x3D
```

```
show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 12 neighbors

Partner's information:

                  LACP port                        Admin  Oper   Port    Port
Port      Flags   Priority  Dev ID          Age    key    Key    Number  State
Gi1/0/1     SP      32768     aabb.cc00.0200   8s    0x0    0xC    0x1     0x3
Gi1/0/2     SP      32768     aabb.cc00.0200  19s    0x0    0xC    0x2     0x3
```

```
show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-------------------------
12     Po12(SU)        LACP      Gi1/0/1(P)    Gi1/0/2(P)
```

# PAgP

## Port Aggregation Protocol (PAgP)

- Only 8 interfaces can be bundled into a PAgP etherchannel
- Unlike LACP, PAgP does support half-duplex interfaces, the only requirement is that the duplex matches on both sides
    - Different duplex interfaces can be combined in the same port-channel, however they must connect to the same value on the other side
- Bundled interfaces need to match on:

    - Speed
    - VLANs
    - Trunk / Access
- Bundled interfaces do not need to match on
    - Duplex
    - STP cost (if hardcoded)
    - Bandwidth (if hardcoded) - Port channel will be torn down at regular intervals because bandwidth does not match
    - PAgP mode (auto or desirable)

- PAgP frames (hello) are sent at a 30 second interval to the L2 multicast address 0100.0ccc.cccc
- Does not support Hot-Standby ports, PAgP port-priority settings have nothing to do with standby interfaces
    - Port-priority and learn methods are used for interoperability between different platforms, not actual traffic distribution

## PAgP Modes

| Mode | Description |
|---|---|
| Desirable | Actively tries to negotiate etherchannel by continuously (30sec) sending PAgP frames |
| Auto | Passively negotiates etherchannel by listening for PAgP frames |

- You can mix and match different modes on interfaces in the same bundle

This is a valid configuration

```
interface range fa1/0/1 - 4
 channel-group 12 mode desirable
 duplex half

interface range fa1/0/5 - 8
 channel-group 12 mode auto
 duplex full
```

**channel-protocol pagp**
**channel-mode** *1-255* **mode** *desirable* | *auto* **non-silent**

- The **channel-protocol** command is not required
- The **non-silent** keyword requires each individual port in the etherchannel to receive a PAgP frame before being added to the channel

- If no PAgP frame is received on the individual port, the port will remain up but will be blocked in STP
- This keyword is recommended to be added when you are sure that there is a PAgP switch on the other end
- Default mode is silent (or not configured)

```
define interface-range PAGP gi1/0/1 - 2

interface range macro PAGP
 shutdown
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 12 mode desirable

interface po12
 switchport trunk encapsulation dot1q
 switchport mode trunk

interface range macro PAGP
 no shutdown
```

```
show pagp internal
Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        d - PAgP is down
Timers: H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.    I - Interface timer is running.

Channel group 12
                              Hello    Partner  PAgP     Learning  Group
Port      Flags State   Timers  Interval Count   Priority   Method  Ifindex
Gi1/0/1   SC    U6/S7   H       30s      1         128        Any      6
Gi1/0/2   SC    U6/S7   H       30s      1         128        Any      6
```

```
show pagp neighbor
Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.        P - Device learns on physical port.

Channel group 12 neighbors
          Partner              Partner          Partner         Partner Group
Port      Name                 Device ID        Port       Age  Flags   Cap.
Gi1/0/1   CSW1                 aabb.cc00.0100   Gi1/0/1    16s  SC      C0001
Gi1/0/2   CSW1                 aabb.cc00.0100   Gi1/0/2    16s  SC      C0001
```

```
show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-------------------------
12     Po12(SU)        PAgP      Gi1/0/1(P)   Gi1/0/2(P)
```

# Static LAG

## Static Link Aggregation Groups (LAG)

- Not recommended
- Can lead to bridging loops because etherchannel is forced to be always on
- May be needed when dealing with VMware environments, otherwise avoid….avoid….avoid

```
interface range gi1/0/1 - 2
 shutdown
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 12 mode on

interface port-channel 12
 switchport trunk encapsulation dot1q
 switchport mode trunk

interface range gi1/0/1 - 2
 no shutdown
```

```
show etherchannel summary

Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------
12     Po12(SU)         -        Gi1/0/1(P)    Gi1/0/1(P)
```

# Link-State Tracking

## Link-State Tracking

- Downstream ports connect to servers (with more than two NICs)
- Upstream ports are part of an etherchannel
- If the etherchannel fails, the downstream ports will be err-disabled triggering a switchover to another NIC on the server

```
link state track 1

int po1
 link state group 1 upstream

int fa0/0
 description SERVER
 link state group 1 downstream

show link state group detail
```

- Ports connected to servers are configured as downstream ports
- Ports connected to other switches are configured as upstream ports
- If the upstream trunk ports (etherchannel) fails, the downstream ports are put in an error-disable state
- This is useful when a server connects to two separate switches with two NICs
- If one switch loses its connection upstream, the NIC port to that switch is shut down and the server will move over to the other switch as the primary NIC port

```
link state track 1
int range fa0/23 - 24
 switchport trunk encapsulation dot1q
 switchport mode trunk
 channel-group 1 mode active
 link state group 1 upstream
```

```
int po1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 link state group 1 upstream

int fa0/0
 switchport mode access
 switchport access vlan 10
 link state group 1 downstream

show link state group detail
```

# LLDP

## Link Layer Discovery Protocol (LLDP)

- Network discovery based on the IEEE 802.1ab standard
- LLDP messages are sent to the multicast address 0180.cc00.0100
- Similar in functionality to CDP, where no actual neighborships are formed
    - Hosts simply advertise their status and receive neighbor status updates
    - Unlike CDP, LLDP can be customized to only send or only receive data independently
- Messages are sent every 30 seconds by default
    - The hold-timer is 4x the advertisement timer (120 seconds by default)
- LLDP carries various pieces of information in Type-Length-Value fields (TLVs)
- LLDP is disabled by default and not supported on all Cisco platforms
    - Enabling LLDP will also enable LLDP-MED

## LLDP-MED (Media Endpoint Discovery)

- Extension to LLDP that operates between endpoint and network devices
- Automatically discovers device type and LAN policies, such as:
    - L2 CoS values
    - Diffserv settings
    - VLANs
    - Power management (PoE classes)
- A network device cannot sent LLDP and LLDP-MED on the same interface
    - By default LLDP will be sent, unless a LLDP-MED TLV is received from the endpoint
    - After reception of a LLDP-MED TLV, the network device will switch to LLDP-MED
    - If the LLDP-MED messages stop, it will resort back to sending LLDP

There are three device classes for LLDP-MED endpoints:

- Class 1 (**generic**) - Basic endpoints like communication controller servers

- Class 2 (**media**) - Endpoints that support streaming, for example media gateways and conference bridges
- Class 3 (**communication device**) - Endpoints supporting IP communications
    - This class is used for VoIP devices such as ip-phones and softphones
    - Most likely class to encounter

## LLDP & LLDP-MED Type-Length-Values (TLVs)

- All LLDP TLVs are enabled by default
- All LLDP-MED TLVs are enabled by default

LLDP Type-Length-Values (TLVs)

| **TLV** | **Definition** |
|---|---|
| System name | The device name |
| Port description | Source port |
| TTL | Time to live (holdtime) |
| Capabilities | Device features |
| System description | IOS version |
| Management address | IP address used for management |
| Port VLAN-id | VLAN present on access port |
| Mac / physical | Duplex mode (half or full)<br>Ethernet Auto-negotiation (speed) |

LLDP-MED Type-Length-Values (TLVs)

| **TLV** | **Definition** |
|---|---|
| System name | The device name |
| Network policy | Advertises VLAN and L3 information<br>A device can be placed in specific VLAN |
| Power management | Advertises power classes, wattage requirements and priority |
| LLDP-MED Capabilities | Device features |

| | |
|---|---|
| Inventory | Device hardware, software, firmware, etc. |
| Location | Physical device location<br>Civic location is address and postcode<br>ELIN location is Emergency Location Identifier Number |
| Port VLAN-id | VLAN present on access port |
| Mac / physical | Duplex mode (half or full)<br>Ethernet Auto-negotiation (speed) |

LLDP packet

```
Link Layer Discovery Protocol
   Chassis Subtype = MAC address, Id: aa:bb:cc:00:01:00
   Port Subtype = Interface name, Id: Et0/0
   Time To Live = 120 sec
   System Name = CSW1
   System Description = Cisco IOS Software
   Port Description = Ethernet0/0
   Capabilities
      0000 111. .... .... = TLV Type: System Capabilities (7)
      .... ...0 0000 0100 = TLV Length: 4
     Capabilities: 0x0014
        .... .... .... ...0 = Other: Not capable
        .... .... .... ..0. = Repeater: Not capable
        .... .... .... .1.. = Bridge: Capable
        .... .... .... 0... = WLAN access point: Not capable
        .... .... ...1 .... = Router: Capable
        .... .... ..0. .... = Telephone: Not capable
        .... .... .0.. .... = DOCSIS cable device: Not capable
        .... .... 0... .... = Station only: Not capable
     Enabled Capabilities: 0x0010
   End of LLDPDU
```

You can customize which information gets sent to neighbors by specifying TLVs

- By default all TLVs are used
- Applies globally for all neighbors or limit certain TLVs per interface

You can disable/enable these TLVs globally:

- 4-wire-power-management
- Mac-phy-cfg
- Management-address
- Port-description
- Port-vlan

- Power-management
- System-capabilities
- System-description
- System-name

no lldp **tlv-select** *tlv_value*

You can disable/enable these TLVs per interface:

- 4-wire-power-management
- Power-management
- LLDP-MED inventory management

interface
 no lldp **tlv-select** *4-wire-power-management* | *power-management*
 no lldp **med-tlv-select** *inventory-management*

## LLDP Configuration

Globally enable LLDP

lldp run

- Disable LLDP globally on all ports with **no lldp run**
- LLDP is disabled by default, enabling LLDP will also enable LLDP-MED
- This command enables both sending (tx) and receiving (rx) of LLDP messages

```
show lldp interface

GigabitEthernet1/0/1:
    Tx: enabled
    Rx: enabled
    Tx state: IDLE
    Rx state: WAIT FOR FRAME
```

Customize LLDP on interface

interface
 no lldp **transmit**
 no lldp **receive**

**transmit** - enable/disable the sending (tx) of LLDP messages
**receive**- enable/disable the reception (rx) of LLDP messages

```
lldp run

interface gi1/0/1
 no lldp transmit
```

```
show lldp interface

GigabitEthernet0/0:
    Tx: disabled
    Rx: enabled
    Tx state: INIT
    Rx state: WAIT FOR FRAME
```

```
show lldp neighbors

Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID           Local Intf     Hold-time  Capability      Port ID
CSW2                Gi1/0/1        120        R               Gi1/0/1
```

Customize LLDP timers

```
lldp timer seconds
lldp holdtime seconds
```

```
lldp timer 20
lldp holdtime 60
```

```
show lldp

Global LLDP Information:
    Status: ACTIVE
    LLDP advertisements are sent every 20 seconds
    LLDP hold time advertised is 60 seconds
    LLDP interface reinitialisation delay is 2 seconds
```

Clear LLDP information

```
clear lldp counters
clear lldp table
```

## LLDP Output

```
show lldp neighbors

Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID           Local Intf     Hold-time  Capability      Port ID
CSW2                Gi1/0/1        120        R               Gi1/0/1
```

- The local interface is the interface on the device
- The port ID is the interface on the neighbor device

The capabilities reflect those in the LLDP packet capture and the output below:

| **Capability** | **Short name** |
|---|---|
| Router | R |
| Station | S |
| DOCSIS Cable Device | C |
| Telephone | T |
| Other | O |
| Repeater | P |
| Access Point | W |

```
show lldp neighbors detail
------------------------------------------------
Local Intf: Gi1/0/1
Chassis id: 00d7.33e3.c800
Port id: Gi1/0/1
Port Description: GigabitEthernet1/0/1
System Name: CSW2

System Description:
Cisco IOS Software
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.

Time remaining: 97 seconds
System Capabilities: B,R
Enabled Capabilities: R
Management Addresses:
    IP: 10.0.0.2
Auto Negotiation - not supported
Physical media capabilities - not advertised
Media Attachment Unit type - not advertised
Vlan ID: - not advertised
```

# Mac Filtering

## MAC Filtering

- Disabled by default and only supports unicast static addresses
- Multicast MAC addresses, broadcast MAC-addresses, and router MAC addresses are not supported (forwarded to the CPU)
- Only works on access-ports, L2 ACL takes precedence over L3 ACL
- You cannot apply named MAC extended ACLs to Layer 3 interfaces
- When filtering the same MAC-address that is also configured statically, the command configured last is applied

```
mac access-list extended MAC_ACL
 deny any any appletalk
 permit any any

int fa0/0
 mac access-group MAC_ACL in

show access-lists
show mac access-group
```

Be careful when only permitting only certain addresses. MAC-addresses will time out and the MAC ACL will block the ARP

- To prevent this, also create static entries that match the ones permitted in the MAC ACL

```
mac access-list extended MAC_ACL
 permit host 1234.abcd.1234 host abcd.1234.abcd
 deny any any

int fa0/1
 mac access-group MAC_ACL in

mac address-table static 1234.abcd.1234 vlan 10 int fa0/0
mac address-table static abcd.1234.abcd vlan 10 int fa0/0
```

# Macros

## Switch Macros

- Some switches come with in macros

    - Show with **show parser macro name** *macro-name* command
- End custom macros with the @ sign
- $ values can be used as variables
- Show the individual commands in the cli when the macro is running with the **trace** keyword

Global macro

```
macro name ACCESS_PORT
 default interface $int
 interface $int
 switchport mode access
 switchport access vlan $vlan
 switchport voice vlan $voice
 no shut
end
@

macro global trace ACCESS_PORT $int fa0/0 $vlan 10 $voice 60
```

Interface macro

```
macro name ACCESS_PORT
 switchport mode access
 switchport access vlan $vlan
 switchport voice vlan $voice

int fa0/0
 macro trace ACCESS_PORT $vlan 10 $voice 60

show parser macro name ACCESS_PORT
```

Define Interface Range

**define interface-range** *macro-name type member/module/number*
**interface range macro** *macro-name*

```
define interface-range RANGE gi1/0/1 - 10 , gi1/0/15 - 16 , gi1/0/20

interface range macro RANGE
 switchport mode access
 switchport access vlan 10
```

# QinQ Tunnel

## QinQ Tunnel

- The native VLAN is not QinQ tagged by default, enforce using the **vlan dot1q tag native** command
- QinQ, like dot1q adds a 4 byte overhead to ethernet frames
- It is possible to send over CDP, STP, and other L2 protocols using QinQ

```
vlan dot1q tag native

int gi0/0
 switchport access vlan 30
 switchport mode dot1q-tunnel
 l2protocol-tunnel cdp
 l2protocol-tunnel dtp

show dot1q-tunnel
```

# SPAN

## Switch-Port Analyzer (SPAN)

- Basically copies the traffic from one port or VLAN (source) to another port (destination)
    - SPAN that copies traffic from a VLAN is called VSPAN
    - SPAN traffic will stay local on the switch (unless you use RSPAN or ERSPAN)
    - Does not affect the switching of traffic on source ports (traffic is still forwarded)
    - The destination port is dedicated for SPAN use and cannot be used for other purposes (unless you use ingress keyword)
    - The destination port line protocol will show as down and the port state is (monitored)
- SPAN will not be operational if the destination port is shutdown
- SPAN does not monitor routed traffic
    - VSPAN only monitors traffic that enters or exits the switch, not traffic that is routed between VLANs

- Source can be multiple ports or VLANs, but not both mixed in the same session
- A maximum of two sessions (local or RSPAN) can exist on the same switch
- Multiple source/destination ports can be combined into a single session
    - An unlimited number of VLANs or source ports can be specified
    - Allows up to 64 span destination ports per switch
- Only a single session per destination port is allowed
    - You cannot send multiple sessions to same port

- Even though STP/CDP/etc. can be copied to the destination port, the port will not participate in STP/CDP
- An etherchannel cannot be the destination of a span session (it can be the source however)

Does not capture traffic that is forwarded within the switch (Inter-VLAN)

- Local SPAN does not copy locally sourced RSPAN and ERSPAN traffic

## SPAN Source

```
monitor session session_number source interface interface_id | vlan vlan_id both | rx | tx
```

**rx** - Receive SPAN (ingress) copies the traffic arriving on the port

- Traffic arriving on the source is copied before modification takes place
  - Modification includes PACLs, QoS markings and policies
  - Traffic is still copied to the SPAN port if the PACL drops the actual traffic

**tx** - Transmit SPAN (egress) copies the traffic arriving on the port

- Traffic arriving on the source is copied after modification takes place
  - Modification includes QoS markings, modified L3 information and policies
  - Traffic is not copied to the SPAN port if the PACL drops the actual traffic

**both** - Combines transmit and receive SPAN (default)

- Captures all traffic that arrives and sends two copies of each packet/frame to the destination
- Traffic arriving on the source is copied twice, before and after modification
- The destination port will receive two copies of the same traffic
- The purpose of this setting is to compare the traffic and verify the effects of markings/policies

Copy traffic twice (ingress & egress) from the source interface

```
monitor session 1 source interface gi1/0/1
```

```
show monitor session 1 detail

 Session 1
 ---------
 Type               : Local Session
 Source Ports       :
     RX Only        : None
     TX Only        : None
     Both           : Gi1/0/1
```

Copy ingress traffic from the source VLAN

```
monitor session 2 source vlan 10 rx
```

```
show monitor session 2 detail

 Session 2
 ---------
 Type               : Local Session
 Source VLANs       :
     RX Only        : None
     TX Only        : 10
     Both           : None
```

## SPAN Destination

monitor session session_number destination interface *interface_id* **encapsulation replicate**

**encapsulation replicate** - Preserve L2 information as it arrived on the source port

- By default, the original L2 encapsulation information is stripped from the packet
  - Traffic is sent untagged, any trunking information is lost
- With encapsulation replicate, the encapsulation is preserved, this includes:
  - Cisco Discovery Protocol (CDP)
  - VLAN Trunk Protocol (VTP
  - Dynamic Trunking Protocol (DTP
  - Spanning Tree Protocol (STP)
  - Port Aggregation Protocol (PAgP)

monitor session ... encapsulation replicate ingress **dot1q vlan** *vlan_id* | **isl** | **untagged vlan** *vlan_id* | **vlan** *vlan_id*

The **ingress** keyword allows traffic from the monitoring station to enter the switch on the SPAN destination port

- This setting allows the monitoring station (or another switch) to still access the network
- The traffic arriving on the destination port will be encapsulation with either dot1q, ISL or a VLAN-id

**ingress dot1q vlan** - Incoming traffic will be encapsulated as 802.1q with the specific VLAN-id
**ingress isl** - Incoming traffic will be encapsulated as ISL
**ingress untagged | vlan** - Incoming traffic will be not be encapsulated and sent untagged with the specific VLAN-id

monitor session 1 destination interface gi1/0/2 encapsulation replicate

```
show monitor session 1 detail

 Session 1
 ---------
 Type               : Local Session
 Destination Ports  : Gi1/0/2
     Encapsulation  : Replicate
```

```
monitor session 2 destination interface gi1/0/2 encapsulation replicate ingress dot1q vlan 99
```

```
show monitor session 2 detail

 Session 2
 ---------
 Type               : Local Session
 Destination Ports  : Gi1/0/2
     Encapsulation  : Replicate
           Ingress  : Enabled, default VLAN = 99
     Ingress encap  : DOT1Q
```

```
show interfaces gi1/0/1

GigabitEthernet1/0/1 is up, line protocol is down (monitored)
```

- The destination port line protocol will show as down and the port state is (monitored)

## SPAN Filtering

- All VLANs active on the trunk are monitored by default
- Use filtering to only capture specific VLANs or subnets (using ACL)
  - Can only be done on trunk or voice VLAN ports
  - Filtering using ACL is called Flow-based SPAN (FSPAN)
- VLAN filtering affects only traffic forwarded to the destination SPAN port
  - Meaning that all traffic is still captured, however only the filtered traffic is sent to the destination

monitor session session_number **filter vlan** *vlan_id/range*

monitor session session_number **filter** *ip* | *mac* | *ipv6* **access-group** *acl*

Only copy certain VLANs to destination port

```
monitor session 1 source interface gi1/0/1
monitor session 1 filter vlan 10,20,100-200
```

```
monitor session 1 destination interface gi1/0/1 encapsulation replicate
```

```
show monitor session 1 detail

Session 1
---------
Type                : Local Session
 Source Ports       :
     RX Only        : None
     TX Only        : None
     Both           : Gi1/0/1
 Source VLANs       :
     RX Only        : None
     TX Only        : None
     Both           : None
 Source RSPAN VLAN  : None
 Destination Ports  : Gi1/0/2
     Encapsulation  : Replicate
           Ingress  : Disabled
  Reflector Port    : None
  Filter VLANs      : 10,20,100-200
  Dest RSPAN VLAN   : None
```

Only copy traffic from sourced by a certain host

```
ip access-list standard HOST1
 permit host 10.0.0.11

monitor session 2 source interface gi1/0/1 rx
monitor session 2 filter ip access-group HOST1
monitor session 2 destination interface gi1/0/1 encapsulation replicate
```

```
show monitor session 2 detail

Session 2
---------
Type                : Local Session
 Source Ports       :
     RX Only        : Gi1/0/1
     TX Only        : None
     Both           : None
 Source VLANs       :
     RX Only        : None
     TX Only        : None
     Both           : None
 Source RSPAN VLAN  : None
 Destination Ports  : Gi1/0/2
     Encapsulation  : Replicate
           Ingress  : Disabled
  Reflector Port    : None
  Filter VLANs      : None
  Dest RSPAN VLAN   : None
  IP Access-group   : HOST1
```

# RSPAN

## Remote Switch-Port Analyzer (RSPAN)

- See SPAN section for general information and limitations
- With RSPAN, the destination 'interface' is not a port but an entire VLAN
  - This VLAN is dedicated for RSPAN usage and cannot be used for other purposes
- The RSPAN-VLAN can be forwarded to another switch
  - In return, this other switch can send the traffic from the RSPAN-VLAN to a destination port
  - RPSAN does not support encapsulation replicate (see SPAN section)
  - RSPAN does support VLAN / ACL filters (see SPAN section)

## RSPAN workings

1. The traffic is sourced from a port on the local switch
2. The traffic is sent to the RSPAN-VLAN and forwarded to a remote switch
3. The remote switch sources the traffic from the RSPAN-VLAN
4. The remote switch sends the traffic to a local destination port

All participating switches must support RSPAN and must have the RSPAN-VLAN configured

## RSPAN-VLAN

- The RSPAN-VLAN must be dedicated with the **remote-span** keyword under the VLAN configuration

    - VTPv3 can propagate RSPAN-VLAN information to other switches (not extended-range)
    - No mac-address learning occurs on the RSPAN-VLAN
    - Cannot be a primary or secondary private-VLAN
- The RSPAN-VLAN only flows over trunk ports, you cannot create an access-port in the RSPAN-VLAN
    - By default the RSPAN-VLAN is active on all trunk ports, so traffic will be flooded on all trunks
    - Either manually prune the RSPAN-VLAN from trunks where it is not needed or use VTP pruning
    - Remember that VTP pruning only works for VLANs 2-1001, so if your RSPAN-VLAN is in the extended-range it will not be pruned

```
vlan vlan_id
 name RSPAN-VLAN
 remote-span
```

**remote-span** - Dedicates the VLAN as an RSPAN-VLAN

```
vlan 999
 remote-span
```

```
show vlan remote-span

 Remote SPAN VLANs
 -----------------
 999
```

Local RPSAN Source

```
monitor session session_number source interface interface_id | vlan vlan_id both | rx | tx
```

**rx** - Receive SPAN (ingress) copies the traffic arriving on the port

- Traffic arriving on the source is copied before modification takes place
    - Modification includes PACLs, QoS markings and policies
    - Traffic is still copied to the SPAN port if the PACL drops the actual traffic

**tx** - Transmit SPAN (egress) copies the traffic arriving on the port

- Traffic arriving on the source is copied after modification takes place
    - Modification includes QoS markings, modified L3 information and policies
    - Traffic is not copied to the SPAN port if the PACL drops the actual traffic

**both** - Combines transmit and receive SPAN (default)

- Captures all traffic that arrives and sends two copies of each packet/frame to the destination

- Traffic arriving on the source is copied twice, before and after modification
- The destination port will receive two copies of the same traffic
- The purpose of this setting is to compare the traffic and verify the effects of markings/policies

Local RPSAN Destination

```
monitor session session_number destination remote vlan vlan_id reflector-port interface_id
```

**remote vlan** - The RSPAN vlan dedicated earlier with the remote-span keyword in the VLAN configuration
**reflector-port** - The resources of this port are 'borrowed' in order to copy traffic to the RSPAN-VLAN

- The reflector port cannot be used for any other purpose, however it must be online and connected
- The reflector port is set to loopback mode and will not be able to send or receive traffic
- Not needed/present on switches that have enough resources available without having to sacrifice a port

Remote RSPAN Source

```
monitor session session_number source remote vlan vlan-id
```

**remote vlan** - The RSPAN vlan dedicated earlier with the remote-span keyword in the VLAN configuration

Remote RSPAN Destination

```
monitor session session_number destination interface interface_id
```

- Though visible in the cli, **encapsulation replicate** is not supported for RSPAN
  - The original VLAN-id is overwritten by the RSPAN-VLAN-id (untagged)
  - See SPAN section for more information about encapsulation replicate
- The destination port also supports the **ingress** keyword
  - This allows traffic from the monitoring station to enter the switch on the SPAN destination port
  - See SPAN section for more information about ingress

## RSPAN Configuration

Local switch (RSPAN Source)

```
vtp version 3
vtp domain cisco
vtp mode server
```

```
vtp pruning

vlan 999
 remote-span

monitor session 1 source interface gi1/0/1 rx
monitor session 1 destination remote vlan 999 reflector-port gi1/0/24
```

```
show monitor session 1 detail

Session 1
---------
Type               : Remote Source Session
 Source Ports      :
     RX Only       : Gi1/0/1
     TX Only       : None
     Both          : None
 Source VLANs      :
     RX Only       : None
     TX Only       : None
     Both          : None
 Source RSPAN VLAN : None
 Destination Ports : None
     Encapsulation : Native
           Ingress : Disabled
  Reflector Port   : Gi1/0/24
  Filter VLANs     : None
  Dest RSPAN VLAN  : 999
```

Remote switch (RSPAN Destination)

```
vtp version 3
vtp domain cisco
vtp mode client

monitor session 1 source remote vlan 999
monitor session 1 destination interface gi1/0/2
```

```
show monitor session 1 detail

Session 1
---------
Type                  : Remote Source Session
 Source Ports         :
     RX Only          : None
     TX Only          : None
     Both             : None
 Source VLANs         :
     RX Only          : None
     TX Only          : None
     Both             : None
 Source RSPAN VLAN    : 999
 Destination Ports    : Gi1/0/2
     Encapsulation    : Native
           Ingress    : Disabled
  Reflector Port      : Gi1/0/24
  Filter VLANs        : None
  Dest RSPAN VLAN     : None
```

# ERSPAN

## Encapsulated Remote Switch-Port Analyzer (ERSPAN)

- Changes only take effect when exiting the sub-configuration mode
  - Sessions also have to be enabled with no shutdown
- The origin ip address keyword specifies the source IP on the ERSPAN source
- The ip address keyword specifies the destination IP on the ERSPAN source
- The ip address keyword specifies the source IP on the ERSPAN destination
- The values for ip address have to match on both sides

Configuring ERSPAN Source

```
monitor session 1 type erspan-source
 source interface gi1
 destination
  erspan-id 12
  ip address 192.168.0.1
  origin ip address 192.168.0.2
  no shutdown
```

Configuring ERSPAN Destination

```
monitor session 1 type erspan-destination
 destination interface gi2
 source
  erspan-id 12
```

```
ip address 192.168.0.1
no shutdown
```

# UDLD

## Protecting Against Sudden Loss of BPDUs

Loopguard and UDLD protect against a sudden loss of BPDUs from a neighboring bridge

- This can be dangerous because switches might (falsely) think that a topology changed has happened
- This behavior can lead to bridging loops when there is no actual TCN and the BPDUs are just delayed / timed out
- UDLD is a better option for etherchannels, because it can block the individual ports

Protects from

- Unidirectional links
- Switches/links that block BPDUs

## Unidirectional Link Detection (UDLD)

- Cisco proprietary protocol
- Exchanges protocol packets that contains the device + port ID and neighbor device + port ID
    - If device does not see its own ID echoed back it considers the link unidirectional
    - Must be enabled on both sides (hello timers do not have to match)
    - Modes also do not have to match on both sides (one can be normal, the other aggressive)
- Initially (before the other side is configured) UDLD will not affect the port state if messages are not echoed back
    - This prevents UDLD from err-disabling ports that connect to workstations that do not speak UDLD

## UDLD Modes

| Mode | Keyword | When unidirectional link is detected |
|---|---|---|
| Normal mode | enable | Port is marked undetermined, and a syslog message is generated<br>Port behaves according to STP state |
| Aggressive mode | aggressive | Tries to re-establish port state a total of 8 times<br>If not successful port is put in errdisable state |

- Aggressive mode can detect:

    - Ports that are stuck (neither transmit/receive but are up)
    - Ports that are up on only one side (fiber)

- UDLD can be enabled globally or on a per port-basis
    - When the feature is enabled globally, it only effects fiber ports
    - The preferred mode for copper links is aggressive

Enable UDLD globally

```
udld enable | aggressive
udld message time seconds (1-90)
```

**enable** - Activates normal mode on all fiber ports
**aggressive** - Activates aggressive mode on all fiber ports
**message time** - Sets the interval at which all UDLD messages are sent

- Default hello interval is 15 seconds for FastEthernet
- Default hello interval is 7 seconds for GigabitEthernet and fiber ports
- Global settings apply to all ports, you cannot configure specific timers for specific ports
- The expiration interval (hold timer) is calculated based on the hello interval

Timers do not have to match between neighbors, all that is needed is that a new hello is received before the expiration time

    - Detection interval (Expiration time) is 45 sec by default
    - The target time must be less than the Max Age timer plus two intervals of the Forward Delay timer, or 50 seconds

Enable UDLD per Interface

```
interface
 udld port aggressive
```

**udld port** - Activates normal mode on specific port
**udld port aggressive** - Activates aggressive mode on specific port

## UDLD Fast Hello

- Provides sub-second unidirectional link detection
    - Hello intervals are between 200 and 1000 msec
- Introduced in IOS 15.0.1 and only supported on a limited number of ports
    - Too many messages can lead to false reports, generally a bad idea to configure
- Both sides must support fast hellos
- Cannot detect unidirectional links when the CPU utilization exceeds 60 percent
- Works similarly to aggressive mode where ports are err-disabled
    - This can be overridden globally (back to enable mode) with the **udld fast-hello error-reporting** command

```
interface
 udld fast-hello interval msec (200-1000)
```

## UDLD Configuration

Normal UDLD in aggressive mode

```
udld message time 5

interface gi1/0/1
 udld aggressive
```

```
show udld neighbors

Port      Device Name   Device ID     Port ID    Neighbor State
----      -----------   ---------     -------    --------------
Gi1/0/1   9R66XVI2CMM   1             Gi1/0/1    Bidirectional
```

```
show udld gi1/0/1

Interface Gi1/0/1
---
Port enable administrative configuration setting: Enabled
Port enable operational state: Enabled
Current bidirectional state: Bidirectional
Current operational state: Advertisement - Single neighbor detected
Message interval: 5000 ms
Time out interval: 5000 ms

Port fast-hello configuration setting: Disabled
Port fast-hello interval: 0 ms
Port fast-hello operational state: Disabled
Neighbor fast-hello configuration setting: Disabled
Neighbor fast-hello interval: Unknown

    Entry 1
    ---
    Expiration time: 43900 ms
    Cache Device index: 1
    Current neighbor state: Bidirectional
    Device ID: 9R66XVI2CMM
    Port ID: Gi1/0/1
    Neighbor echo 1 device: 9KP1ROC199K
    Neighbor echo 1 port: Gi1/0/1

    TLV Message interval: 15 sec
    No TLV fast-hello interval
    TLV Time out interval: 5
    TLV CDP Device name: CSW1
```

Fast UDLD in enable mode

```
udld fast-hello error-reporting

interface gi1/0/1
 udld fast-hello interval 500
```

```
show udld fast-hello

 Total ports on which fast hello can be configured: 16
 Total ports with fast hello configured: 1
 Total ports with fast hello operational: 1
 Total ports with fast hello non-operational: 0
 Port-ID    Hello Neighbor-Hello Neighbor-Device Neighbor-Port Status
 -------    ----- -------------- --------------- ------------- ------
 Gi1/0/1    500   500            9KP1ROC199K     Gi1/0/1       Operational
```

Re-enable ports

```
udld reset
```

**reset** - Reset all interfaces which have been err-disabled by UDLD

Or

```
errdisable recovery cause udld
```

```
show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
udld                         Enabled

Timer interval: 300 seconds
```

# Switch Design / Cabling

# Errors & Mismatching

## Errdisable Recovery

- Default errdisable recovery interval is 300 seconds (5 min)
- By default a switch will not automatically recover any errdisable cause
  - This is because **errdisable recovery cause** is not configured
  - This command is different from **errdisable detect cause**, which specifies the events can be detected
  - The **errdisable recovery interval** is configured at 300 seconds by default
- Errdisable detection cause is enabled by default on all types
  - Errdisable recovery cause is disabled by default on all types
- Settings (including recovery interval) are applied globally for all ports / reasons on the switch

**errdisable detect cause** *all* | *cause-name*

```
show errdisable detect
ErrDisable Reason            Detection        Mode
-----------------            ---------        ----
arp-inspection               Enabled          port
bpduguard                    Enabled          port
channel-misconfig (STP)      Enabled          port
community-limit              Enabled          port
dhcp-rate-limit              Enabled          port
dtp-flap                     Enabled          port
ekey                         Enabled          port
gbic-invalid                 Enabled          port
iif-reg-failure              Enabled          port
inline-power                 Enabled          port
invalid-policy               Enabled          port
l2ptguard                    Enabled          port
link-flap                    Enabled          port
link-monitor-failure         Enabled          port
loopback                     Enabled          port
lsgroup                      Enabled          port
oam-remote-failure           Enabled          port
mac-limit                    Enabled          port
pagp-flap                    Enabled          port
port-mode-failure            Enabled          port
pppoe-ia-rate-limit          Enabled          port
psecure-violation            Enabled          port
security-violation           Enabled          port
sfp-config-mismatch          Enabled          port
storm-control                Enabled          port
udld                         Enabled          port
unicast-flood                Enabled          port
vmps                         Enabled          port
psp                          Enabled          port
dual-active-recovery         Enabled          port
```

errdisable recovery **cause** *all* | *cause-name*
errdisable recovery **interval** *seconds*

errdisable recovery cause link-flap
errdisable recovery interval 30

```
show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
arp-inspection               Disabled
bpduguard                    Disabled
channel-misconfig (STP)      Disabled
dhcp-rate-limit              Disabled
dtp-flap                     Disabled
gbic-invalid                 Disabled
inline-power                 Disabled
l2ptguard                    Disabled
link-flap                    Disabled
mac-limit                    Disabled
link-monitor-failure         Disabled
loopback                     Disabled
oam-remote-failure           Disabled
pagp-flap                    Disabled
port-mode-failure            Disabled
pppoe-ia-rate-limit          Disabled
psecure-violation            Enabled
security-violation           Disabled
sfp-config-mismatch          Disabled
storm-control                Disabled
udld                         Disabled
unicast-flood                Disabled
vmps                         Disabled
psp                          Disabled
dual-active-recovery         Disabled

Timer interval: 30 seconds

Interfaces that will be enabled at the next timeout:

Interface       Errdisable reason       Time left(sec)
---------       -----------------       --------------
Gi1/0/1         link-flap                 29
```

```
show interfaces status err-disabled

Port      Name               Status       Reason               Err-disabled VI
Gi1/0/1                      err-disabled link-flap
```

## Speed & Duplex Mismatches

- Runts are packets that were truncated before they were fully received

- Runts and input errors on the link are an indicator that port settings are mismatched

```
show interfaces gi1/0/1
GigabitEthernet1/0/1 is up, line protocol is up (connected)
  Hardware is AmdP2, address is aabb.cc00.0120 (bia aabb.cc00.0120)
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed, media type is unknown
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
     2085 packets input, 438424 bytes, 0 no buffer
     Received 2045 broadcasts (0 multicasts)
     31235 runts, 0 giants, 0 throttles
     123346 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     147424321263 packets output, 1213123595364 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

# Ethernet Cabling

## Naming Conventions

- T = Twisted pair copper cable
- F = Optical cable (Fiber)
- L = Long range single- or multi-mode fiber
- S = Short range single- or multi-mode fiber
- C = Balanced copper cable
- X = Coding method

## Speed Types

**10 Mb/s (IEEE 802.3)**

- 10BASE-T

- Half-duplex

**100 Mb/s (IEEE 802.3u)**
- Uses only 4 wires (2 pairs), exception is 100BASE-T4
- Cat5 cabling
- 100BASE-TX / 100BASE-FX / 100BASE-T2/T4
- Half-duplex and full-duplex

**1000 Mb/s (IEEE 802.3z)**
- Uses all 8 wires (4 pairs)
- Cat5e or Cat6 cabling
- 1000BASE-T / 1000BASE-SX / 1000BASE-LX
- Half-duplex (not implemented) and full-duplex

**10 Gb/s (IEEE 802.3au)**
- Cat6 cabling only
- Full-duplex only
- 10GBASE-SR/SW / 10GBASE-LR/LW

## Crossover and Straight-Through
- Network devices are MDIX and end-devices MDI
    - Crossover cables alternate the receiving (rx) and transmit (rx) pairs
    - This is needed on devices that do not support auto-MDIX
    - GigabitEthernet uses all 4 pairs, so using crossover is not recommended (or even usable),
    - Rely on auto-MDIX instead for Gigabit (and FastEthernet)

# Hierarchical Design

## Hierarchical Network
- A L2 switch is a transparent bridge, or a multiport transparent bridge

**(A) Access Layer (L2)**
- High port density / low cost per port
- Highly available / scalable uplinks to Distribution
- Port security / QoS

**(D)Distribution Layer (L3)**
- Aggregation of Access
- High L3 throughput
- Security and access policies
- QoS

- High port density of high speed links
- Scalable / redundant links to Access & Core

VLANs and broadcast domains converge here and L3 filtering/routing/policies are applied
Switches used here must be capable of high throughput routing and are a L3 boundary between VLANs
Do not extend L2 beyond the D layer
D layer should always be the boundary of VLANs, subnets and broadcast traffic

**(C) Core Layer (L3)**

- Very high L3 throughput
- Plain forwarding, no policies or filtering
- High port density of high speed ports
- Efficient switches that forward even when at 100% capacity
- Redundancy and advanced QoS

A collapsed core design combines the D and C layers into one

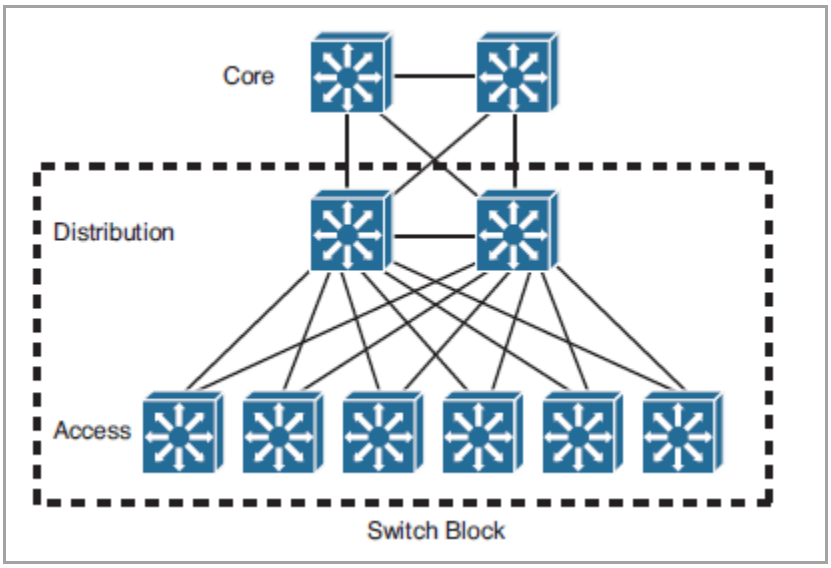
## Modular Design

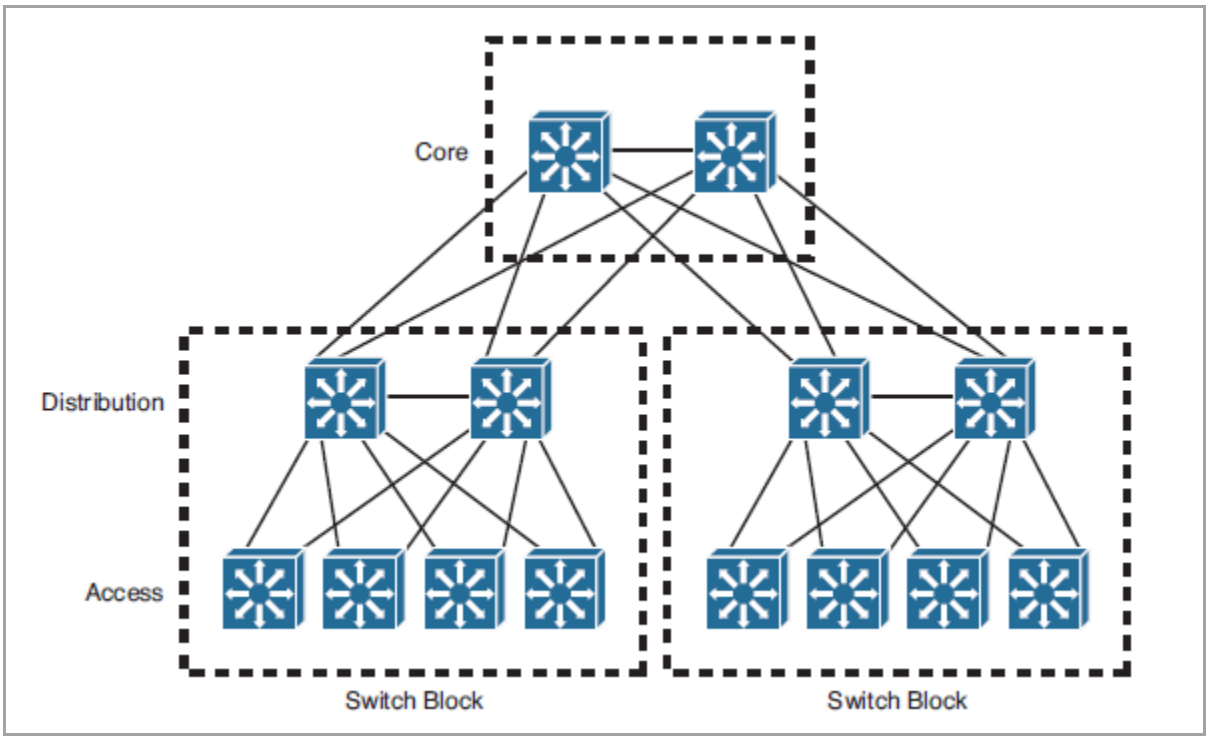
- Linking all A switches to all D switches can quickly become disorganized
- The solution is to create switch blocks

**Switch Block**

- A group of A and D switches
- Also called an access distribution block
- The C layer connects all the switch blocks
- Links between D and C switches should preferably be L3
- A dual core or redundant core is a C layer of two identical switches
    - A multi-node core is a C layer of four identical switches

A switch block will reduce overall redundancy but will create a more logical designed network that is organized




- The size of a switch block is dependent on traffic-types, behavior and size of workstations/VLANs
- The amount of D switches in a switch block depends on the amount of A switches / port density needed
    - The number of D switches is usually 2

If a switch block is too large the D layer becomes a bottleneck due to the large amount of inter-VLAN traffic / policies / security checks

- Another issue is the propagation of broadcast and multicast traffic to many destinations

## VLAN Design

- Ideally, you should not allow VLANs to extend beyond the Layer 2 domain of a distribution switch
- The VLAN should stay inside a switch block and NOT reach across a network's core and into another switch block
- The idea again is to keep broadcasts and unnecessary traffic movement out of the core block

VLANs can be scaled in the switch block by using two basic methods:

- End-to-end VLANs (campus-wide VLANs)
- Local VLANs

The 80-20 and 20-80 Rule

- 80-20 = 80% of users traffic is local, 20% is for remote resources
- 20-80 = 20% of users traffic is local, 80% is for remote resources

End-to-end VLANs

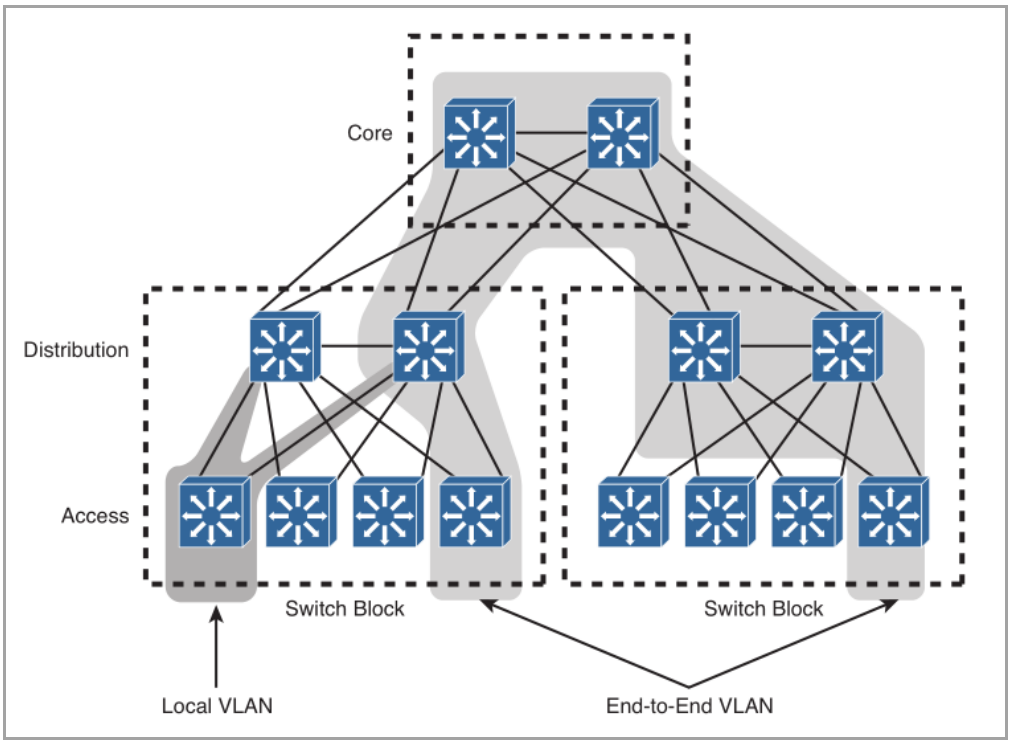
- Span the entire switching network
- Offers Flexibility and mobility where users can plug in their device in any access switch
- Follows the 80-20 rule
- Older-style design where many resources (servers/applications) were available in the local network
- Easy management of users

Characteristics:

- End-to-end VLANs are present on core switches
- Users are grouped into a VLAN based on function, not location
- Users can plug in their device in any access switch and be assigned their respective VLAN (requires VMPS/dot1x/etc.)

Issues:

- End-to-end VLANs are present on core switches, leading to an increase in broadcast traffic
- Possibility of bridging loops
- Large STP domain and many blocking links
- Does not scale well
- Complex troubleshooting




Local VLANs

- Only present on the local switch
- Users cannot receive the same VLAN when they plug in at different parts of the network
    - However resources are usually located outside of the network, so this does not matter as much
- Follows the 20-80 rule
- Newer-style design where many resources (servers/applications) are available outside the local network

Characteristics:

- VLANs are not present on core switches
- Users are grouped into a VLAN based on location, not function
- Scales well
- Easy troubleshooting
- Resources are accessed outside of the local network
- Layer 3 links between switches
- Almost non-existent STP domain
- No possibility of bridging loops
- Decreased broadcast traffic

Issues:

- Possible security issues if users are no longer segmented
- May need to configure many PACLs/ACLs in order to achieve segmentation

# MAC / (T)CAM

## Flooding & Learning

L2 traffic is divided into 3 categories:

- Unknown unicast
  - Flooded on all ports except the port it was received on
  - Mac-address of source is learned and stored in CAM table (mac-address table)
  - After successful communication between two hosts, the switch should have dynamically learned two source mac-addresses and associated VLANs + interfaces
  - Example traffic is when a frame is received with source + destination but it is not known to the switch (not in the mac-address table)
- Unknown multicast
  - Flooded on all ports except the port it was received on
- Unknown broadcast
  - Always flooded on all ports except the port it was received on
  - Example traffic types is ARP or Gratuitous ARP

Learning

- All interfaces and VLANs can learn MAC-addresses
- It is possible to disable learning for specific VLANs

```
no mac address-table learning vlan 10
show mac address-table learning
```

Gratuitous ARP (ARP requests)

- Sent by host when interface comes up to check its own ip-address
  - The host does not expect a reply to this request
  - When a reply is received the host knows that its ip-address is in use elsewhere in the network
- Also used to update ARP tables after a mac-address for a known IP changes
  - Or a known mac-address is now on a different port (host is re-plugged)
- Also used by FHRP to update MAC tables on L2 devices with the virtual MAC address

## Drop Unknown Unicast/Multicast Traffic

- Block flooding of unknown multicast and unicast traffic out of a port
- The default is to flood unknown traffic out of all ports, except the port it was received on
- Blocking this traffic can lead to the breaking of major switching features
  - When mac-address time out, hosts will have no method of responding to arp-requests
  - Use with caution, preferably combine with static mac-address entries

interface
**switchport block** *multicast* | *unicast*

## Content-Addressable Memory (CAM) Table

- Mac-address table
- Contains learned source mac-addresses, VLAN IDs and egress switch ports
  - Address are learned dynamically and will age out (default) in 300 seconds
  - It is possible to apply different aging timers for different VLANs
- Binary table, addresses either match exactly (1) or not (0)
- If the same mac-address is learned on a different port, the existing entry is purged from the table
  - This is to prevent duplicate mac entries when hosts are re-plugged to another port
  - If a mac-address keeps being learned on alternating ports, a flapping error message will be generated
- This process is silent and will not generate a syslog message, unless you configure **mac-address-table notification mac-move** globally
  - This command causes the switch to generate a syslog when a mac-address changes ports
  - It doesn't generate a syslog when a mac-address is added or removed from the CAM table

mac address-table **aging-time** *seconds* **vlan** *vlan-id*
debug platform matm learning

mac address-table aging-time 300
mac address-table aging-time 100 vlan 10

```
show mac address-table aging-time
Global Aging Time:  300
Vlan    Aging Time
----    ----------
  10     100
```

```
show mac address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0000.0000.0001    DYNAMIC     Gi10/1
   1    0000.0000.0002    DYNAMIC     Gi10/2
   1    0000.0000.0003    DYNAMIC     Gi10/3
Total Mac Addresses for this criterion: 3
```

## Static MAC

- Does not expire and can be associated with multiple ports
- The specified output interface cannot be an SVI
- When configuring static MAC-address in a primary private-VLAN, also configure the same address in the secondary VLAN

mac address-table **static** *mac-address* **vlan** *vlan-id* **interface** *interface_id*

```
mac address-table static 0000.0000.0001 vlan 1 interface gi1/0/1
```

```
show mac address-table static
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    0000.0000.0001    STATIC      Gi1/0/1
Total Mac Addresses for this criterion: 1
```

## Ternary Content-Addressable Memory (TCAM) Table

- Used for filtering (ACL), QoS policies and forwarding decisions
  - ACLs are made up access control entities (ACEs) that are evaluated in sequential order
  - ACLs and their entries are evaluated simultaneously or parallel with a L2/L3 forwarding decisions
- Consulted before the CAM table
- Contains 134bit Value, Mask, and Result (VMR) combinations
  - Values consist of source and destination IPs, protocols and ports
  - Masks mark the important value bits and decide whether they matter or not
  - Results are the decision on what to do with the traffic (drop due to ACL entry, or remark QoS etc.)

- Consists of two components:
  - Feature Manager (FM). Compiles the ACEs into entries in the TCAM table
  - Switching Database Manager (SDM). Tunes the TCAM table to optimize its resources based on templates

The amount of masks and values supported in TCAM depend on the switch hardware and the SDM template

Display masks/values in use

```
show platform tcam utilization
 CAM Utilization for ASIC# 0 Max Used
 Masks/Values Masks/values
 Unicast mac addresses: 6364/6364 31/31
 IPv4 IGMP groups + multicast routes: 1120/1120 1/1
 IPv4 unicast directly-connected routes: 6144/6144 4/4
 IPv4 unicast indirectly-connected routes: 2048/2048 2047/2047
 IPv4 policy based routing aces: 452/452 12/12
 IPv4 qos aces: 512/512 21/21
 IPv4 security aces: 964/964 30/30

 Note: Allocation of TCAM entries per feature uses
 a complex algorithm. The above information is meant
 to provide an abstract view of the current TCAM utilization
```

# PoE

## Power over Ethernet Modes & Standards

| PoE mode / standard | Naming | Output | Proprietary |
|---|---|---|---|
| Cisco Inline Power | ILP | 7W | Yes |
| IEEE 802.3af | PoE | 15.4W | No |
| IEEE 802.3at | PoE+ | 25.5W | No |
| Cisco Universal PoE | UPoE | 60W | Yes |
| IEEE 802.3bt | PoE++ | 96W | No |

- Cisco Inline Power (ILP) is a proprietary method that was developed before the IEEE standards (Around 2000)
  - ILP only supports BASE10/100 (Ethernet + FastEthernet) cabling
- The first standardized form of PoE was in 2003 (802.3af by the IEEE)
- PoE+ was standardized in 2009 (802.3at by the IEEE)

- PoE++ is an upcoming standard supposed to be standardized somewhere in 2017 (802.3bt by the IEEE)

Terminology
- Power Sourcing Equipment (PSE) - A network device (switch) that provides the power
- Powered Device (PD) - A network device (phone / access-point / etc.) that requires power

## PoE Types & Modes

| PoE type | Description | Standard | Proprietary |
|---|---|---|---|
| Active | The PSE and PD negotiate the voltage that the PD needs<br>The PD is placed in a power class (see below) | Yes | Unlikely |
| Passive | The PSE always outputs a fixed voltage | No | Most likely |

- The passive/active naming convention is confusing because it implies that passive PoE is not always on
- The naming relates to the negotiation of PoE:
  - Active PoE devices negotiate the amount of power (voltage) that is required
  - Passive PoE devices do not negotiate anything

Two modes of powering devices are used:
- Mode A - The power is delivered on the data pairs of the cable
  - This is made possible due to phantom power
- Mode B - The power is delivered on the spare pairs of the cable

GigabitEthernet (unlike FastEthernet) uses all 4 pairs of the cable. However it is still compatible with mode B
- The spare pairs in this case will be the pairs that would be spare if the cable was running at 10/100 speed

## Power Class Discovery (Detection)
- Active PoE allows the PSE and PD to negotiate the voltage/wattage required
  - Negotiation is performed through CDP or LLDP
- Based on the requirements, the PD is placed into a PoE class

| Class | Usage | Power Range | Max Power |
|---|---|---|---|
| 0 | Default | 0.44–12.95W | 15.4W |
| 1 | Optional | 0.44–3.84W | 4W |
| 2 | Optional | 3.84–6.49W | 7W |

| | | | |
|---|---|---|---|
| 3 | Optional | 6.49–12.95W | 15.4W |
| 4 | 802.3at | 12.95–25.50W | 30W |
| - | UPoE | - | 60W |

- Default class (0) is used if the powered device does not support power class discovery and provides 15.4W
  - This is usually the default power output provided by 802.3af
- If a device requires more power (30W) it will inform the switch trough LLDP or CDP (Requires 802.3at support)

## PoE Configuration

```
interface
 power inline auto | never | static max max_wattage (milliwatts)
```

**auto** - The PD can request the amount of power needed through CDP or LLDP
- The PSE will deliver power up to the maximum of 30W (as long as there is power available and the PSE is not oversubscribed)
- You can customize the maximum amount of power with the max keyword

**static** - The PSE will deliver a fixed amount of (maximum) wattage and no negotiation is performed (basically passive PoE)
- If the PD requires less power than is delivered, no adjustment is made
- If the PD requires more power than is deliver, the port is shutdown

**never** - The PSE disables PoE on the port and power is never delivered

**max** - Sets the maximum wattage (in milliwatts) on the specific port
- The range is 4000 to 30000 mW
- The default value is the maximum (30000 mW or 30 W)

```
interface range gi1/0/1 - 12
 power inline auto max 25500

interface range gi1/0/13 - 24
 power inline static max 15400
```

- Ports gi1/0/1-12 will negotiate the PoE settings / wattage and will be limited to 25.5 W maximum
- Ports gi1/0/13 - 24 will not negotiate the PoE settings / wattage and will be limited to 15.4 W maximum

Fast PoE

```
interface
 power inline port poe-ha
```

- The PSE remembers the last known power negotiation on the port
- After the PSE recovers after a reboot (or power-failure) it will immediately provide the last known wattage on the port
    - This allows devices to boot up faster after a power failure, but could also crash the PSE again in case of oversubscription

Police PoE

```
interface
 power inline police action log | errdisable
```

**log** - Generate a syslog message if the wattage exceeds the maximum power allocation on the port (but continue providing power)
**errdisable** (default) - Shuts down the port if the wattage exceeds the maximum power allocation on the port (30 W by default)

- You can automatically recover errdisabled ports with the **errdisable recovery cause inline-power** global command
    - The errdisable detection of PoE events is on by default (**errdisable detect cause inline-power**) and the recovery interval is 300 seconds by default

```
show power inline interface [ detail ]
```

```
show power inline
Module  Available     Used      Remaining
        (Watts)      (Watts)    (Watts)
------  ---------   --------    ---------
1         710.0       110.4       599.6

Interface   Admin   Oper    Power   Device      Class   Max
                                                        (Watts)
--------------------------------------------------------------
Gi1/0/1     auto    on  16.8    CAP3702I-A      4       30.0
Gi1/0/2     auto    on  6.3     IP Phone 7912   n/a     30.0
Gi1/0/3     auto    off 0.0     n/a             n/a     30.0
Gi1/0/4     auto    off 0.0     n/a             n/a     30.0
Gi1/0/5     auto    on  6.3     IP Phone 7910   n/a     30.0
Gi1/0/6     auto    off 0.0     n/a             n/a     30.0
Gi1/0/7     auto    on  6.3     IP Phone 7910   n/a     30.0
Gi1/0/8     auto    on  4.0     IEEE PD         1       30.0
Gi1/0/9     auto    on  4.0     IEEE PD         1       30.0
Gi1/0/10    auto    on  6.3     IP Phone 7942   2       30.0
```

```
show power inline gigabitethernet1/0/1 detail
 Interface: Gi1/0/1
 Inline Power Mode: auto
 Operational status: on
 Device Detected: no
 Device Type: cisco AIR-CAP3702I-
 IEEE Class: 4
 Discovery mechanism used/configured: Unknown
 Police: off
 Power Allocated
 Admin Value: 30.0
 Power drawn from the source: 16.8
 Power available to the device: 16.8

 Actual consumption
 Measured at the port: 6.2
 Maximum Power drawn by the device since powered on: 9.2

 Absent Counter: 0
 Over Current Counter: 0
 Short Current Counter: 0
 Invalid Signature Counter: 0
 Power Denied Counter: 0
```

- Leaving the max power value at the default of 30W on all ports can lead to issues

- The maximum available wattage of the above switch is 710W
    - If this was a 48-port switch and all ports are used for PoE 30W it would require 1440W, roughly double of what it can provide
- Configure a max wattage on each port separately, or disable some ports for inline power to make sure the switch can never be oversubscribed

# SDM Templates

## Switching Database Manager (SDM) Templates

- Configure system resources in the switch to optimize support for specific features
- Different templates allocate resources in a different manner
- Requires reload in order to apply new template

The most common templates

| SDM Template | Usage | Result |
|---|---|---|
| Access | Maximize system resources for ACLs | More access control entries (ACE)<br>Less unicast routes and mac-addresses |
| Default (Desktop) | Balanced | Balanced resource allocation |
| Dual-IPv4-and-IPv6 | Supports IPv6 routing (disabled by default) | More unicast routes for IPv4 and IPv6<br>Less overall mac-addresses and ACEs |
| Routing | Optimize switch for L3 routing | More unicast routes<br>Less mac-addresses |
| VLAN | Optimized for access switches<br>Supports maximum number of mac-addresses<br>Supports large number of clients | Unicast routing is disabled<br>0 unicast routes<br>More mac-addresses |

```
sdm prefer access | vlan | routing | default
```

Enable both IPv4 and IPv6 services

```
sdm prefer dual-ipv4-and-ipv6 access | vlan | routing
```

Revert back to default

```
no sdm prefer
```

```
show sdm prefer
 The current template is "desktop routing" template.
 The selected template optimizes the resources in
 the switch to support this level of features for
 8 routed interfaces and 1024 VLANs.

  number of unicast mac addresses:            3K
  number of igmp groups + multicast routes:   1K
  number of unicast routes:                   11K
    number of directly connected hosts:       3K
    number of indirect routes:                8K
  number of qos aces:                         0.5K
 number of security aces:                     1K

 On next reload, template will be "desktop vlan" template.
```

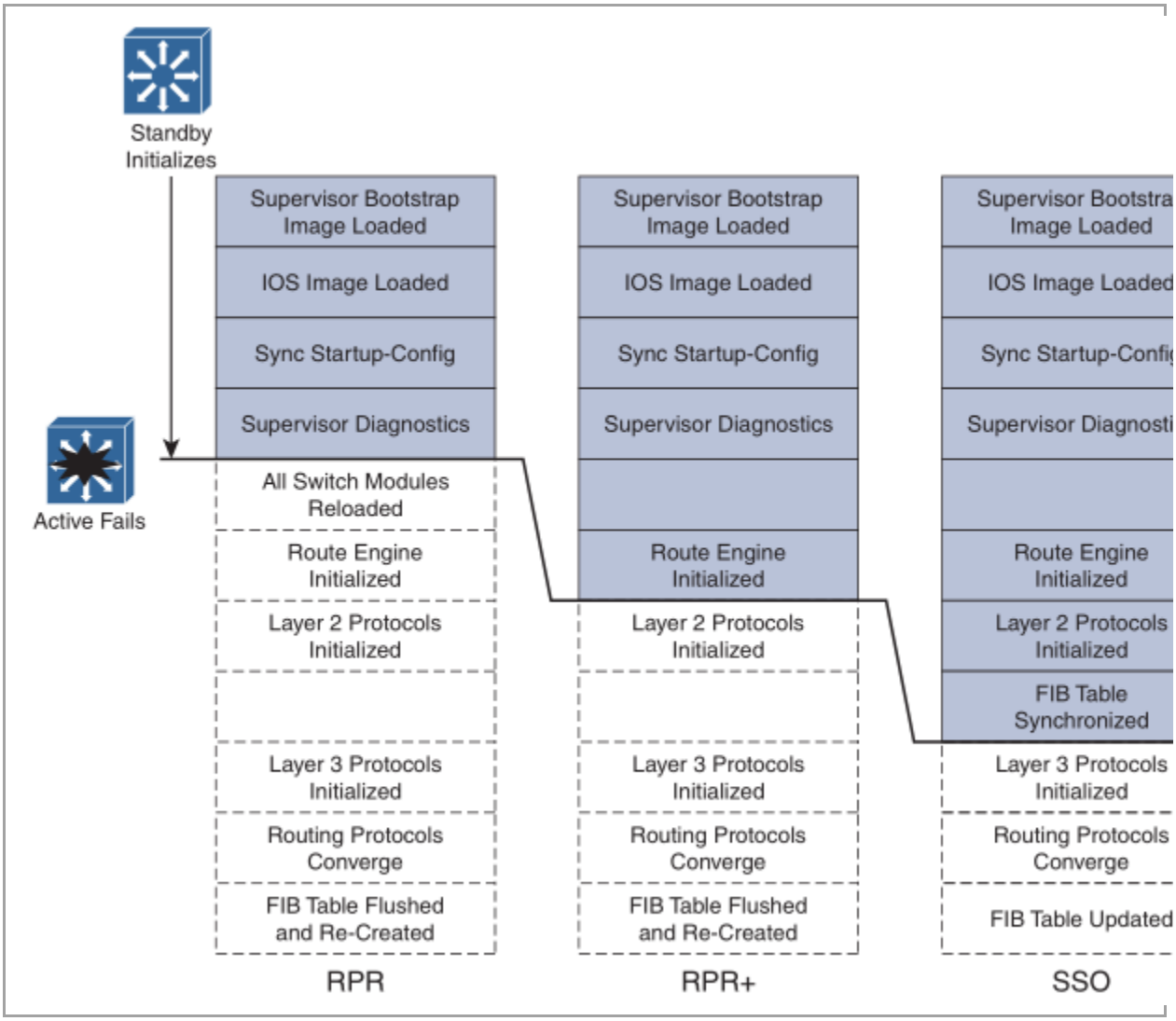
# Stacking

## Redundancy Modes

| Mode | Name | Redundant supervisor | In case of failure | Failover time |
|---|---|---|---|---|
| RPR | Route Processor Redundancy | Partially booted<br>Partially initialized | Standby must reload every other module<br>Standby must initialize all supervisor functions | > 2 minutes |
| RPR+ | Route Processor Redundancy Plus | Booted<br>Supervisor initialized<br>Route engine initialized | Standby finishes initializing<br>Standby starts L2 and L3 functions | > 30 seconds |
| SSO | Stateful Switchover | Fully booted<br>Supervisor initialized<br>Route engine initialized<br>Configs synced<br>L2 synced | Standby immediately takes over<br>L2 is already synced<br>L3 functions are started (see NSF below) | > 1 second |

RPR+ requires that the images match exactly between members

- If the images do not match, the stack will fall back to using RPR




redundancy
**mode** *rpr* | *rpr-plus* | *sso*

## Non-Stop Forwarding (NSF)

- Quickly rebuilds the L3 information in case of an active/standby switchover
  - The RIB is rebuild on the other member after a supervisor switchover
  - The RIB can then be used to quickly rebuild the FIB
- All members need to support NSF
- Implemented in routing protocols only, not on static routes
  - Needs to be turned on for each specific protocol

Supported protocols:

- BGP
- EIGRP
- OSPF
- IS-IS

router bgp *as-number*
**bgp graceful-restart**

router eigrp *as-number*
**nsf**

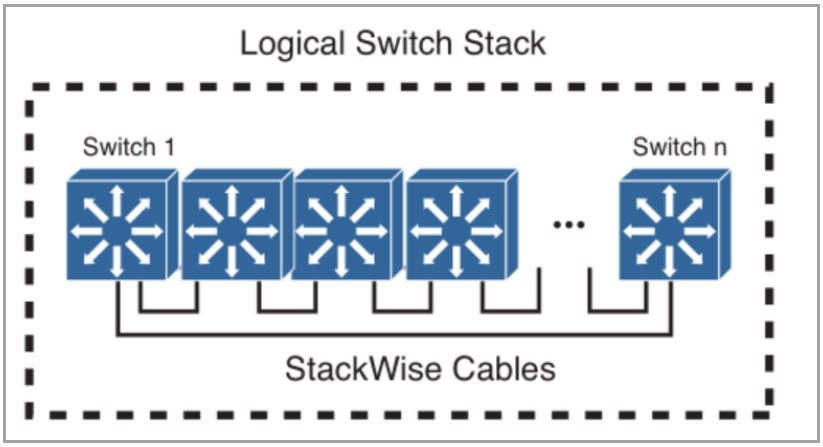
router ospf *process-id*
**nsf**

router isis *tag*
**nsf** *cisco* | *ietf*

# StackWise

## StackWise

- StackWise uses a closed stacking loop (stacking ring)
  - All switches are connected in series and the last switch in the chain connects back to the first switch
  - This design makes it possible to add and remove switches from the stack, without interruption
  - Removes the need for a FHRP protocol



## StackWise Master & Slaves

One of the switches in the stack will become the stack master

- This switch holds the control plane and management functionality
- All the other switches become slaves and only forward frames
    - Depending on the platform, up to nine switches (including the master) can be part of a stack
    - The master is indicated with a * in **show switch** (see below)

When a switch is added to the stack, it will receive the config and image from the master

- The mac-address table of the new switch is also updated to reflect all the other members
- Each switch is aware of all other switchports and mac-address tables

## Master Election & Configuration

1. Priority - Highest priority switch will become the master, range is 1-15 (default is 1 for all switches)
2. Image Type - Most extensive feature set (IP Services will win over IP base image)
3. Default Configuration - Pre-configured will win over default-configuration switch
4. Uptime - The switch with the longest uptime will win
5. Mac-address - The switch with the lowest mac-address will win

If you power on two identical switches with default configuration at the same time, the decision will probably fall back on the lowest mac-address

| **switch** *stack-member-number* **priority** *1-15* |
|---|

Changing the priority has no effect on a converged stack

- This is not preemptive, the master will only change when:
    - The switch stack is reset
    - The stack master is removed from the switch stack
    - The stack master is reset or powered off
    - The stack master fails
    - The switch stack membership is changed (adding / removing members)

| **stack-mac persistent timer** *0-60 min* |
|---|

Time delay after a stack-master change before the stack MAC address changes to that of the new master

- Default timer is 4 minutes
- If you configure 0, you disable the mac-address change (old master mac-address will be used indefinitely)
    - This will not change until manual intervention with **no stack-mac persistent timer**

**switch** *stack-member-number* **renumber** *1-8*

Changes the unit-number of a stack member after convergence

```
stack-mac persistent timer 7
switch 1 priority 15
```

```
show switch
Switch/Stack Mac Address : 0016.4727.a900
Mac persistency wait time: 7 mins
                                           H/W   Current
Switch#  Role   Mac Address     Priority Version  State
-----------------------------------------------------------
*1       Master 0016.4727.a900     15       0     Ready
 2       Member 0018.123a.0b3a     1        0     Ready
```

```
show switch stack-ports summary
Sw#/Port#  Port Status  Neighbor  Cable Length   Link OK   Link Active   Sync
-------------------------------------------------------------------------------
1/1        OK           2         50cm           Yes       Yes           Yes
1/2        OK           2         50cm           Yes       Yes           Yes
2/1        OK           1         50cm           Yes       Yes           Yes
2/2        OK           1         50cm           Yes       Yes           Yes
```

## Stack Cabling

Cisco uses special proprietary cables for stacking (Cisco StackWise Technology Resilient Cabling)

- Supports up to 16 Gbps in both directions (32 Gbps bi-directionally), depending on platform
- When switches are connected in a ring, the most optimal path is chosen to the neighbor switch
    - This is done through QoS, which prefers the link with the lowest bandwidth load

```
show switch stack-ring speed

Stack Ring Speed        : 32G
Stack Ring Configuration: Full
Stack Ring Protocol     : StackWise
```

# VSS

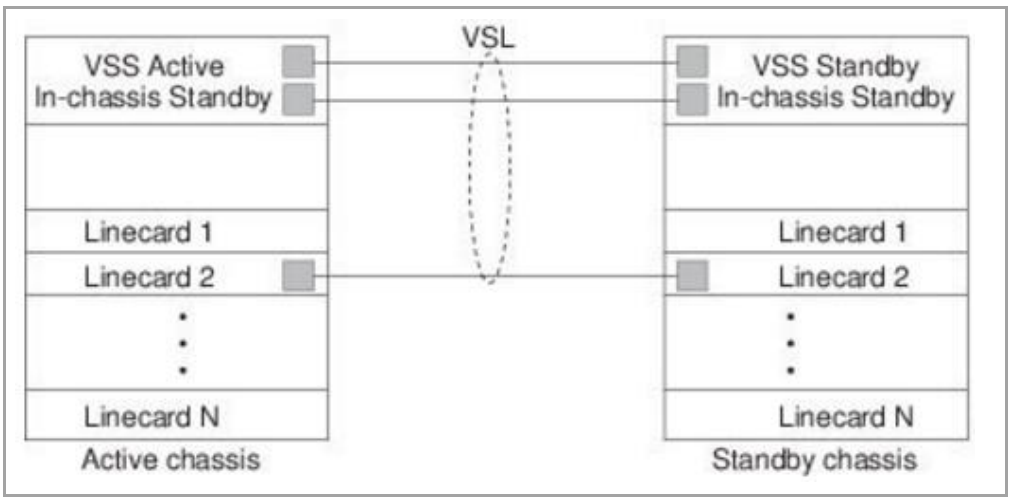
## Virtual Switching System (VSS)

Combines two (only two) physical switches into a logical switch (stack)

- VSS is supported on platforms such as the catalyst 4500/6500
- Removes the need for a FHRP protocol
- One switch will be active and the other will be standby

Virtual Switch Link (VSL) is an etherchannel between two chassis with up to 8 links

- Requires the usage of two port-channel (Po) interfaces (1 for each chassis, remember they are combined)
    - Supports only 10 Gb/s ethernet ports
    - 8x 10-Gigabit Etherchannel (10GEC) = throughput of up to 160 Gbps (2 directions)
    - Supports LACP / PAgP and static LAGs
- Carries control and data traffic between active and passive chassis
    - Control traffic has a higher priority than data so that they are never lost
    - Data traffic is load-balanced using the port-channel load-balancing method
- All traffic that flows over the VSL is encapsulated with a special 32-byte header (so it is very similar to ISL)
    - Traffic needs to traverse the VSL if it is flooded, or a packet destination is only plugged into one of the two chassis




A multi-chassis Etherchannel (MEC) is a LAG divided between multiple members of a stack/VSS (see Etherchannel section)

- In this case, the VSS can function as a core switch and distribution switches can connect to both chassis at the same time
- The VSS is smart enough to detect which chassis should be used to forward traffic (if say the host wants to reach a device that is only plugged into one of the two chassis)

## Active and Passive Chassis

| Chassis | Function |
|---|---|

| | |
|---|---|
| Active | Controls the VSS,<br>Provides management functions<br>Online insertion and Removal (OIR) + console<br>Switches L2 and L3<br>Packet forwarding on local interfaces<br>Runs STP instance |
| Standby | Packet forwarding on local interfaces<br>Sends management traffic to active chassis |

- Active/passive chassis is based on switch number and priority
  - Switch 1 will be active if priority is equal (100 by default)
  - Switch 2 will be active if priority is lower than switch 1
  - Does not preempt unless chassis is reloaded or if you force it with **redundancy force-switchover**
- If there is a VSS failover, the mac-address used by STP will remain the same (the previously active chassis mac-address)

Virtual Switch Link Protocol (VSLP)
Contains the Link Management Protocol (LMP) and Role Resolution Protocol (RRP)

- LMP runs on all VSL links and exchanges information required to establish communication between the two chassis
  - After a LMP connection is established, the peer chassis start using RRP
- RRP is used to negotiate the role (active or passive) for each chassis

VSS & SSO Redundancy
With Stateful Switchover (SSO) redundancy, the VSS standby supervisor engine is always ready to take over the active role

- The standby chassis assumes control following a fault on the VSS active supervisor engine

VSS Initiation process

1. Preparse config
2. Bring up VSL links
3. Run VSLP (LMP)
4. Run RRP
5. Interchassis SSO
6. Continue system bootup

## VSS Configuration

Set the redundancy mode to Stateful Switchover(SSO)

```
redundancy
 mode sso
```

Set the router protocol (if used) to use NSF

```
router routing_protocol
 nsf
```

Define the switch domain and switch number

```
switch virtual domain domain_number (1-255)
 switch switch_number (1-2) priority priority_value (1-255)
```

**virtual domain** - Basically defines the two chassis as neighbors and must match on both sides
**switch** - Identifies the switch in the chassis and must be unique

- The chassis with number 1 will become the active switch, and number 2 will become passive
    - The above is true, unless you configure the switch priority as well

**priority** - Lower priority is better, a chassis with number 2 with a lower priority will become active

- This change can only happen on a reload of the VSS or if you enter **redundancy force-switchover**
- VSS does not preempt if you configure a new priority, unless you force it with the command above

Configure the VSL on chassis #1

```
interface port-channel port_channel_1
 switch virtual link switch_number
```

Configure the VSL on chassis #2

```
interface port-channel port_channel_2
 switch virtual link switch_number
```

**switch virtual link** - defines the chassis number (created in the virtual domain above) to be the owner of this port-channel

- Both active and passive chassis need to use a DIFFERENT port-channel interface

Add ports to the etherchannel on chassis #1

```
interface range tengigabitethernet
 channel-group port_channel_1 mode on | active | desirable
```

Add ports to the etherchannel on chassis #2

```
interface range tengigabitethernet
 channel-group port_channel_2 mode on | passive | auto
```

Convert chassis #1 to VSS

```
switch convert mode virtual
```

Convert chassis #2 to VSS

```
switch convert mode virtual
```

- This is a privilege exec mode command
- Go back to standalone chassis, with the **switch convert mode stand-alone** privilege exec command

CSW1

```
redundancy
 mode sso

router ospf 1
 nsf

switch virtual-domain 12
 switch 1 priority 100

interface po10
 switch virtual link 1

interface range te1/0/1 - 2
 channel-group 10 mode on

switch convert mode virtual
reload
```

CSW2

```
redundancy
 mode sso

router ospf 1
 nsf

switch virtual-domain 12
 switch 2 priority 100

interface po20
 switch virtual link 2
```

```
interface range te2/0/1 - 2
 channel-group 20 mode on

switch convert mode virtual
reload
```

```
show switch virtual role

 Switch  Switch Status  Preempt    Priority  Role     Session ID
         Number         Oper(Conf) Oper(Conf)         Local  Remote
 ------------------------------------------------------------------
 LOCAL    1     UP      FALSE(N)   100(200)  ACTIVE   0      0
 REMOTE   2     UP      FALSE(N)   100(100)  STANDBY  8158   1991

 In dual-active recovery mode: No
```

```
show switch virtual link
 VSL Status : UP
 VSL Uptime : 1 day, 3 hours, 39 minutes
 VSL SCP Ping : Pass
 VSL ICC Ping : Pass
 VSL Control Link : Te 1/0/1
```

```
show switch virtual link port-channel

 Group  Port-channel  Protocol    Ports
 ------+-------------+-----------+----------------------
 10     Po10(RU)         -        Te1/0/1(P) Te1/0/1(P)
 20     Po20(RU)         -        Te2/0/1(P) Te2/0/2(P)
```

## VSS Dual Active Recovery Detection

Problems can occur if both chassis are online but the VSL between them fails

- In this case both chassis will think they are active and will assume the active role
- A VSL failure with both chassis online will create a dual-active scenario and can lead to reachability issues
    - For example, both chassis will think they are the root and will use the same mac-address + priority
    - Both chassis will use the same ip-address for management (ssh/https connections)

In order to prevent this, the switches must be able to detect a dual-active scenario

- This detection is achieved by using communication methods outside of the VSL link
- For example, the communication between the active/standby chassis can be done through other switches that connect to the VSS
- There are three methods to detect a dual-active scenario (all three communicate via other switches/links):

    - Enhanced PAgP - Both chassis communicate with each other via PAgP over the MEC links to neighboring (distribution) switches
    - BFD - Connect a separate ethernet link between the two chassis which will run Bidirectional Forwarding Detection (BFD)
    - Dual-Active Fast Hello - Sends hello-messages over a separate ethernet link between the two chassis (similar to BFD, but faster)
- You can configure all methods at the same time
    - Cisco recommends to use at least two ports for the backup ethernet links (using one or more of the above methods)

<u>During Recovery</u>

- If the VSL connection goes down, the active switch will be informed over the backup connection and will go into recovery mode
    - In this mode, all ports on the active chassis except the VSL ports are shut down (err-disabled)
    - The switch will wait until the VSL comes back online
    - If it detects that the VSL is back online, the switch will reload and come back as the standby switch with all ports forwarding
- The standby switch will initiate SSO and will become the active switch
    - All ports on the new active switch will be placed in the forwarding mode

Shutting down all ports in case of recovery may lead to unwanted results

- For example, certain ports may be used for management or other L3 services
- You can exclude ports from being err-disabled by configuring exclusion lists

```
switch virtual domain 12
 dual-active exclude interface gi1/0/48
```

```
show switch virtual dual-active summary

 Pagp dual-active detection enabled: Yes
 Bfd dual-active detection enabled: Yes
 Fast-hello dual-active detection enabled: Yes
 Interface Gi1/0/48 excluded from shutdown in recovery mode
```

Another action you can take is creating a backup management address (recovery ip) for the active switch

- When the active switch is in recovery mode, the standby (new active) will receive connections for the management ip
- Configure a recovery ip to still manage the previous active switch

```
switch virtual domain 12
 dual-active recovery ip address 10.0.0.199 255.255.255.255
```

## VSS Dual Active Recovery with Enhanced PAgP

- Requires the operation of PAgP on the MEC between the VSS and other (distribution) switches
  - The neighbor switch must also support enhanced PAgP
- PAgP messages include a TLV which contains the ID of the VSS active switch.
  - Only switches in VSS mode send the new TLV
  - The PAgP message is sent through the neighboring switch (via the MEC) and arrives on the other chassis member
- Enabled by default on PAgP trusted port-channels (**dual-active detection pagp**)
  - However no port-channels are designated as trusted by default
  - The port-channel has to be administratively down before it can be trusted

```
interface range gi1/0/3 , gi2/0/3
 description CSW3
 channel-group 3 mode desirable

interface po3
 shutdown

switch virtual domain 12
 dual-active detection pagp
 dual-active detection pagp trust channel-group 3

interface po3
 no shutdown
```

```
show switch virtual dual-active pagp

 PAgP dual-active detection enabled: Yes
 PAgP dual-active version: 1.1

 Channel group 3 dual-active detect capability w/nbrs Dual-Active trusted grou
           Dual-Active     Partner              Partner    Partner
 Port      Detect Capable  Name                 Port       Version
 Gi1/0/3   Yes             CSW3                 Gi1/0/1    1.1
 Gi2/0/3   Yes             CSW3                 Gi1/0/2    1.1
```

The **show pagp dual-active** command displays the same information as **show switch virtual dual-active pagp**

## VSS Dual Active Recovery with BFD

- Uses a separate ethernet link between the two chassis which will run Bidirectional Forwarding Detection (BFD)

    - These interfaces must be directly connected to both chassis
- BFD is used because it does not rely on the management ip-address
    - Both chassis share the same ip-address so an icmp-echo with ip sla for example will not work
- Requires pre-configured ports with an ip-address and relevant BFD configuration

```
interface gi1/0/24
 no switchport
 ip address 169.254.12.1 255.255.255.252
 bfd interval 500 min_rx 500 multiplier 3

interface gi2/0/24
 no switchport
 ip address 169.254.21.1 255.255.255.252
 bfd interval 500 min_rx 500 multiplier 3

switch virtual domain 12
 dual-active detection bfd
 dual-active pair interface gi1/0/24 interface gi2/0/24
```

```
show switch virtual dual-active bfd

 Bfd dual-active detection enabled: Yes
 Bfd dual-active interface pairs configured:
   interface1 Gi1/0/24 interface2 Gi2/0/24
```

## VSS Dual Active Recovery with Fast Hello

- Uses a separate ethernet link between the two chassis which will run dual-active fast hellos
    - These interfaces must be directly connected to both chassis
    - A maximum of four interfaces is supported
- Does not rely on any form of ip-addresses
- Enabled globally by default (**dual-active detection fast-hello**) but not on interfaces
    - All existing configuration is removed when you enable fast hello on an interface

```
interface gi1/0/23
 dual-active fast-hello
interface gi2/0/23
 dual-active fast-hello

switch virtual domain 12
 dual-active detection fast-hello
```

```
show switch virtual dual-active fast-hello
 Fast-hello dual-active detection enabled: Yes

 Fast-hello dual-active interfaces:
 Port       State (local only)
 ---------------------------
 Gi1/0/23   -
 Gi2/0/23   -
```

# Switch Security

# 802.1X (Port-Authentication)

## Port-Based Authentication (802.1X / dot1x)

- Traffic is only forwarded after user has authenticated to the switch (aaa server)
- 802.1x uses Extensible Authentication Protocol over LANs (EAPoL)
    - 802.1x can be combined with DHCP snooping option 82 to insert dot1x information into the DHCP packet
- 802.1x requires the client to supply a username and password, verified by a RADIUS server
- Only RADIUS authentication (in combination with dot1x) is supported on IOS
    - Multiple RADIUS servers can be defined to service requests
    - Servers are consulted in the order in which they were configured (see aaa radius section)
- Only a single host is expected per switch-port (default)
    - Override this behavior using the **dot1x host-mode multi-host interface** command
    - This setting is useful when a switch connects to another access-switch with clients

Both client (host) and switch need to support 802.1x

- If the switch does not support 802.1x, traffic will be sent normally
- If the client does not support 802.1x, switchport will remain unauthorized and traffic will be blocked

When a client connects, initially only 802.1x (l2) traffic is allowed between the switch and client

- Only after authentication is successful can other protocols start communicating
    - This includes ip-addressing information (DHCP), which will be blocked
- Both the client and the switch can initiate the session
    - The authenticated session ends when the client sends a termination request, or the port is timed out
- Radius servers do not speak EAPoL (they do support EAP) and clients do not understand RADIUS
    - The switch acts as a translation device (relay) between EAPoL and RADIUS

- The ethernet header (oL) is stripped and the EAP frame is encapsulated in a radius packet

## 802.1X Device Roles / Terminology

| Device role | Name | Description |
|---|---|---|
| Supplicant | Client / host | Requests access to the LAN by supplying credentials<br>Must support 802.1X-compliant client software |
| Authenticator | Switch | Relays EAP messages between supplicant and the authentication server<br>Enables/disables ports based on the success/failure of authentication<br>Must support radius and EAPoL |
| Authentication Server | Radius server | Stores credentials and informs the authenticator whether client is accepted or denied |

If 802.1x authentication times out the switch can fallback to another authentication method

- If configured, the switch can fallback to:
  - Mac-Authentication Bypass (MAB) - Switch relays client mac-address to the server for authorization
  - Web-Based Authentication (WEBAUTH) - Switch presents client with a splash page (HTTP) asking for credentials
- If not configured, the request will time-out and the client is never authenticated

interface
**dot1x pae** *authenticator* | *supplicant* | *both*

The switch can act as an authenticator (default) or as a dot1x client (supplicant), or both at the same time

- PAE stands for Port Access Entity

## 802.1X Host-Modes

| Host modes | Description | # of allowed mac-addresses |
|---|---|---|
| Single-host (default) | Only one client can be connected to the 802.1X-enabled port | 1 |
| Multi-host | Allow multiple hosts on an 802.1x-authorized port after the directly connected host has been authenticated<br>Useful when the device connected is an access-point | * |

| | | |
|---|---|---|
| | Clients associated with the access-points do not have to authenticate to the switch<br>Clients can still authenticate to the AP if it supports dot1x | |
| Multi-domain | Allow both a host and a VoIP phone to be connected to the 802.1X-enabled port<br>Useful when client desktops connect to the network through ip-phones | 2 |
| Multi-auth | Allow one client on the voice VLAN and multiple authenticated clients on the data VLAN<br>Only available with the **authentication host-mode** command<br>Useful when the device connected is an access-point<br>Unlike multi-host, it requires each connected client to individually authenticate | * |

interface
**authentication host-mode** *multi-auth* | *multi-domain* | *multi-host* | *single-host*

interface
**dot1x host-mode** *single-host* | *multi-host* | *multi-domain*

## 802.1X Violation Modes

| Mode | Port Action | Administrative Action |
|---|---|---|
| Protect | Traffic from violating hosts is dropped<br>No record of the violation is kept | Port will automatically resume after violations stop |
| Restrict | Traffic from violating hosts is dropped<br>Record of the violation is kept<br>Switch can send a SNMP trap and/or syslog message | Port will automatically resume after violations stop |
| Shutdown (default) | Port is error-disabled (shutdown) | Port must be re-enabled manually (shut / no shut)<br>Port will automatically resume if **err-disable recovery** is configured |
| Replace | Removes the current session and authenticates with the new host | Port will change to authenticate new host |

- You can configure err-disable recovery to automatically bring up disabled ports with the **errdisable recovery cause security-violation** global command

```
interface
 authentication violation shutdown | replace | protect | restrict
```

## 802.1X Global Configuration

- Like port-security, dot1x is only supported on statically defined ports (access / trunk)
    - Not supported on dynamic ports (DTP)

- The **dot1x….** commands are the older style of configuration commands
- The **authentication….** commands are the newer style

```
aaa new-model
```

- Globally enable AAA functionality on the switch

```
radius-server host hostname | ip-address key-string
```

- You can also define custom radius groups (see aaa radius section)
- Each server is polled in the order in which they were configured

```
aaa authentication dot1x default group radius
```

```
dot1x system-auth-control
```

This enables 802.1x globally on the switch

- The default authentication port-state is force-authorized (see below)
- This state permits all switchports to forward traffic regardless of authentication
    - Default state is permit any any

## 802.1X Port State Configuration

| 802.1X port state | Description |
|---|---|
| Force-Authorized (default) | Client is always authorized to send traffic (default) |
| Force-Unauthorized | Client is never authorized to send traffic (even after successful authentication) |
| Auto | 802.1x decides whether client is authorized or not to send traffic |

```
interface
 dot1x port-control force-authorized | force-unauthorized | auto
```

```
interface
 authentication port-control force-authorized | force-unauthorized | auto
```

## 802.1X Re-Authentication

```
interface
 authentication periodic
 authentication timer reauthenticate | inactivity 1-65535 seconds
```

**reauthenticate** - Absolute timer
**inactivity** - Starts counting down after traffic from host stops

```
interface
 dot1x reauthentication
 dot1x timeout reauth-period seconds
```

- This enables periodic reauthentication of the client
  - By default the client is authenticated until the session is terminated (indefinitely)
- You can manually (force) client reauthentication with the **dot1x re-authenticate interface** privilege command
  - Clear dot1x authentication information (sessions) with the **clear dot1x** *all* | *interface* privilege command

## 802.1X Guest-VLAN

- The switch can put non-speaking clients into a special purpose VLAN
  - Used to provide access to clients that do not support dot1x (do not respond to switch queries)
  - Guest-VLAN is not implemented by default
- All VLANs are supported except:
  - RSPAN VLAN
  - Primary private-VLAN
  - Voice VLAN
- Not supported on trunk ports, only on access

```
interface
 authentication event no-response action authorize vlan vlan-id
```

```
interface
 dot1x guest-vlan vlan-id
```

## 802.1X Restricted-VLAN

- The switch can put unauthenticated clients (who failed authentication) into a special purpose VLAN
  - Used to provide access to guest-clients that do not have credentials in order to provide limited connectivity
  - You can set the failed authentication attempts needed before a client is placed in the restricted-VLAN
  - Restricted-VLAN is not implemented by default
- A port in the restricted VLAN tries to reauthenticate the client at configured intervals
  - Reauthentication (for restricted VLAN, not normal) is enabled by default on a 60 second timer
  - If disabled, the only way client can reauthenticate is by a link-down event (unplugging/re-plugging the host machine)
- All VLANs are supported except:
  - RSPAN VLAN
  - Voice VLAN
- Not supported on trunk ports, only on access

```
interface
 authentication event fail retry 0-5 retries action authorize vlan vlan-id
```

- Default is 2 retries

```
interface
 dot1x auth-fail vlan-id
 dot1x auth-fail max-attempts 0-3 attempts
```

- Default is 3 attempts

## Newer 802.1X Configuration

- Uses **authentication** interface level commands

Place hosts in VLAN 100 if they do not support dot1x

```
aaa new-model
radius-server host 1.1.1.1 cisco
aaa authentication dot1x default group radius
dot1x system-auth-control
```

```
interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 authentication port-control auto
 authentication host-mode single-host

 authentication event no-response action authorize vlan 100
```

```
show dot1x interface gi1/0/1

Dot1x Info for GigabitEthernet1/0/1
-----------------------------------
PAE = AUTHENTICATOR
PortControl = AUTO
ControlDirection = Both
HostMode = SINGLE_HOST
QuietPeriod = 60
ServerTimeout = 30
SuppTimeout = 30
ReAuthMax = 2
MaxReq = 2
TxPeriod = 3
```

Place hosts in VLAN 100 if they fail dot1x authentication

```
aaa new-model
radius-server host 1.1.1.1 cisco
aaa authentication dot1x default group radius
dot1x system-auth-control

interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 authentication port-control auto
 authentication event fail retry 2 action authorize vlan 100
 authentication host-mode multi-auth

 authentication periodic
 authentication timer reauthenticate 3600
```

- An access-point is connected to interface gi1/0/1
- All wireless clients are authenticated
    - If they fail two tries they will be placed in the restricted VLAN 100
    - Every 60 seconds they will try to re-authenticate and join VLAN 10

- After they have authenticated (joined VLAN 10) they will be re-polled every hour to verify status

## Older 802.1X Configuration

- Uses **dot1x** interface level commands

Place hosts in VLAN 100 if they fail dot1x authentication

```
aaa new-model
radius-server host 1.1.1.1 cisco
aaa authentication dot1x default group radius
dot1x system-auth-control

interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 dot1x port-control auto
 dot1x pae authenticator
 dot1x host-mode single-host

 dot1x auth-fail 100
```

Place hosts in VLAN 100 if they do not support dot1x

```
aaa new-model
radius-server host 1.1.1.1 cisco
aaa authentication dot1x default group radius
dot1x system-auth-control

interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 switchport voice vlan 60
 dot1x port-control auto
 dot1x pae authenticator
 dot1x host-mode multi-domain
 dot1x guest-vlan 100
 dot1x reauthentication
 dot1x timeout reauth-period 3600
```

- A phone and computer are connected to interface gi1/0/1
- Both the phone and computer are authenticated
  - If either do not support dot1x, they will be placed in VLAN 100

    - After they have authenticated (joined VLAN 10 + VLAN 60) they will be re-polled every hour to verify status

# DHCP Snooping

## DHCP Snooping

- DCHP snooping inspects DHCP traffic to make sure no rogue switches can be installed on the network
- An access/trunk port connected to a switch can either be untrusted (default) or trusted
    - Only trust ports where DHCP servers are located
    - Untrusted ports can be rate-limited, to limit the amount of DHCP messages can be sent per second

- The switch basically acts as a DCHP relay, inspecting and inserting options into the packet
    - Any packets that have the gateway address (giaddr) set will be dropped, unless the port is trusted for relaying (see below)
    - If the DHCPDISCOVER originated on the host, the giaddr is 0.0.0.0, this will not be altered by the snooping switch
    - The switch also inserts the snooping information option 82

- When this DHCPDISCOVER arrives at the DHCP server, it will arrive appearing to be relayed (option 82) but will not have a set giaddr
    - On Cisco IOS DHCP servers, this is not a valid packet and will be dropped (other platforms may behave differently)

```
debug ip dhcp server packet
DHCPD: relay information option exists, but giaddr is zero
```

Three methods to trust snooped/relayed packets on the DHCP server:

- Disable the insertion of the information option 82 on the snooping switch (**no ip dhcp snooping information option**)
- Trust relayed DHCP packets on the server with a giaddr of 0.0.0.0 and information option 82
    - Configure **ip dhcp relay information trusted** on interface facing the switch (or VLAN)
    - Or, configure **ip dhcp relay information trust-all** globally

## DHCP Option 82

- DHCP option 82 (relay information option) identifies hosts by both the mac-address and the switchport
    - Insertion is enabled by default on snooping switches, disabled by default on DHCP servers/relays

    - It allows DHCP relays to inform the DHCP server where the original request came from

There are two ways the information option 82 is inserted in a DHCP packet:

| ip dhcp relay information option |
|---|

- Applies to DHCP relays
- Disabled by default

| ip dhcp snooping information option |
|---|

- Applies to DHCP snooping switches
- Enabled by default

## DCHP Snooping Binding Database

- Contains information about untrusted hosts with leased IP addresses, holds:
    - Mac-address of the host
    - Leased ip-address of the host
    - Lease time
    - Binding type
    - VLAN-id
    - Interface host is connected to
- Dynamic ARP inspection (DAI) and IP Source Guard also use information stored in the DHCP snooping binding database
- View the binding database with **show ip dhcp snooping binding**
- Clear the database with **clear ip dhcp snooping binding**

- By default the information in this table is stored in volatile memory, meaning it will be cleared on a reboot
    - It may be preferable to store the database in a persistent storage location (remote or local)

| **ip dhcp snooping database** *location* **write-delay** *seconds* **timeout** *seconds* |
|---|

**write-delay** - The entries are batched together and written in one action, the delay specifies how long the switch waits before writing (300 sec default)
**timeout** - If the write action fails to complete the transfer in this period, it will be aborted (300 sec default)

## Configuring DHCP Snooping

- You need to configure both these options to enable DHCP snooping:

- Enable snooping globally with **ip dhcp snooping**
- Configure the operational snooping for specific VLAN with **ip dhcp snooping** *vlan-id*

By default, DHCP snooping will verify:

- Gateway address (giaddr field)
- Hardware address (hwaddr field)

Override with

**no ip dhcp snooping verify** *mac-address* | *no-relay-agent-address*

Snooping Switch

```
ip dhcp snooping
ip dhcp snooping vlan 10
no ip dhcp snooping information option

interface gi1/0/1
 description DHCP_SERVER
 ip dhcp snooping trust

interface gi1/0/2
 description DHCP_CLIENT
 switchport mode access vlan 10
 ip dhcp snooping limit rate 10
```

```
show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10
DHCP snooping is operational on following VLANs:
10
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is disabled
   circuit-id default format: vlan-mod-port
   remote-id: aabb.cc00.0100 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                  Trusted    Allow option    Rate limit (pps)
-----------------------    -------    ------------    ----------------
GigabitEthernet1/0/1       yes        yes             unlimited
GigabitEthernet1/0/2       no         no              10
```

```
show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interfac
------------------  ---------------  ----------  -------------  ----  --------
00:50:79:66:68:00   10.0.0.11        86297       dhcp-snooping   10   GigabitE
Total number of bindings: 1
```

```
debug ip dhcp snooping packet
```

```
show ip dhcp snooping statistics
 Packets Forwarded                                    = 18
 Packets Dropped                                      = 0
 Packets Dropped From untrusted ports                 = 0
```

DHCP server

```
interface vlan 10
 ip dhcp relay information trusted
```

Or

```
ip dhcp relay information trust-all
```

- Configure these options when not disabling the **ip dhcp snooping information option** on the switch

```
show ip dhcp relay information trusted-sources
List of trusted sources of relay agent information option:
Vlan10
```

## DHCP Snooping + Relay

- When using multiple DHCP Snooping devices untrusted relay information must be allowed on the switch closest to the server
    - Alternatively DHCP option 82 can be disabled on the switch not connected to the server (furthest away from server)

Closest to clients (CSW1)

```
vlan 10
vlan 172

service dhcp
ip dhcp snooping
ip dhcp snooping vlan 10
no ip dhcp snooping information option

interface gi1/0/1
 description TO_CSW2
 switchport mode trunk

interface gi1/0/2
 description DHCP_CLIENT
 switchport mode access vlan 10

interface vlan 10
 description DHCP_RELAY
 ip address 10.0.0.1 255.255.255.0
 ip helper-address 172.16.0.100

interface vlan 172
 ip address 172.16.0.1 255.255.255.0
```

Closest to server (CSW2)

```
vlan 172

ip dhcp snooping
```

```
ip dhcp snooping vlan 172

interface gi1/0/1
 description TO_CSW1
 switchport mode trunk

interface gi1/0/2
 description DHCP_SERVER
 switchport mode access vlan 172
 ip dhcp snooping trust

interface vlan 172
 ip address 172.16.0.2 255.255.255.0
 ip dhcp relay information trusted
```

# ARP Inspection (DAI)

## Dynamic Arp Inspection (DAI)

- Validates ARP packets using source ip/mac-address bindings (can be customized)
  - These IP-to-MAC bindings are stored in the DHCP snooping database
  - Intercepts ARP replies and compares the information to the known bindings
  - Drops invalid ARP reply traffic
  - When the arp packets per second rate-limit is exceeded (see below) the port is err-disabled
- Requires the presence of DHCP snooping and uses the snooping database by default
  - Alternatively use static entries with an ARP access-list
- Only untrusted interfaces (all by default) will be validated
  - Use **ip arp inspection trust** to trust an interface (not inspect the traffic on that interface)
  - Usually, interfaces connected to other switches will be trusted
- Untrusted interfaces are rate-limited to 15 arp packets per second
  - Customize with the **ip arp inspection limit** interface command
  - You can also rate-limit trusted ports, default is unlimited pps
- Only works at the ingress level on L2 interfaces (and only for ARP replies)
- Optionally recover err-disable violating ports with **errdisable recovery cause arp-inspection** command

## ARP Verification

- By default only ip/mac-address bindings in the ARP reply are verified
  - Basically, only the ARP message is inspected which is encapsulated in a packet > frame
  - The extra options also verify the information present in the packet and frame

Can verify one or more of the following:

- Destination mac-address in frame containing ARP reply
- Source mac-address in frame containing ARP reply
- Destination ip-address in packet containing ARP reply
- Source ip-address in packet containing ARP reply

```
ip arp inspection validate dst-mac | ip | src-mac
```

**ip** - Compare the ip information in the ARP reply against the information in the IP packet
**src-mac** - Compare the source mac-address in the ARP reply against the one in the Ethernet header
**dst-mac** - Compare the destination mac-address in the ARP reply against the one in the Ethernet header

```
show ip arp inspection
 Source Mac Validation      : Enabled
 Destination Mac Validation : Enabled
 IP Address Validation      : Enabled
```

## ARP Rate-Limiting

```
ip arp inspection limit rate pps burst interval seconds
ip arp inspection limit none
```

**rate** - Max mount of incoming arp packets that are processed per second, range is 0-2048 (default is 15)
**burst interval** - The interval at which the port is monitored for high rates of ARP packets (default is 1)
**none** - Set the max amount to unlimited

## DAI Configuration

Switch CSW1

```
ip dhcp snooping vlan 10
ip arp inspection vlan 10

interface gi1/0/1
 description CSW2
 switchport mode trunk
 ip arp inspection trust
```

```
interface gi1/0/2
 description CLIENT
 switchport access vlan 10
 ip arp inspection limit rate 10 burst-interval 2
```

```
show ip arp inspection
Vlan     Configuration    Operation   ACL Match          Static ACL
----     -------------    ---------   ---------          ----------
  10     Enabled          Active

Vlan     ACL Logging      DHCP Logging
----     -----------      ------------
  10     Deny             Deny
```

```
show ip arp inspection interfaces
Interface        Trust State     Rate (pps)    Burst Interval
---------------  -----------     ----------    --------------
Gi1/0/1           Trusted              None
Gi1/0/2           Untrusted              10                 2
```

```
show ip arp inspection statistics
Vlan      Forwarded        Dropped     DHCP Drops     ACL Drops
----      ---------        -------     ----------     ----------
  10              2              0              0             0
Vlan   DHCP Permits    ACL Permits   Source MAC Failures
----   ------------    -----------   -------------------
  10              2              0                     0
Vlan   Dest MAC Failures   IP Validation Failures
----   -----------------   ----------------------
  10                  0                        0
```

## DAI with ARP Access-List

- ARP access-lists are checked before the snooping database
  - Can also be configured without DHCP snooping enabled
  - If there is no match in the ACL, the switch will check the snooping database

**arp access-list** *arp-acl-name*
 **permit ip host** *ip-address* **mac host** *mac-address* **log**

**ip arp inspection filter** *arp-acl-name* **vlan** *vlan-id* **static**

**static** - Do not check the snooping database after no match is found

- You can also configure an explicit deny in the ACL

```
arp access-list STATIC_HOSTS
 permit ip host 10.0.0.101 mac host 0000.0000.0001 log

ip arp inspection filter STATIC_HOSTS vlan 10
```

```
show ip arp inspection
Vlan      Configuration    Operation   ACL Match          Static ACL
----      -------------    ---------   ---------          ----------
  10      Enabled          Active      STATIC_HOSTS       No

Vlan      ACL Logging      DHCP Logging
----      -----------      ------------
  10      Deny             Deny
```

# IP Source Guard

## IP Source Guard

- Checks the source IP address of received packets against the DHCP snooping binding database
- If the source address does not match an entry in the binding database, it will be dropped
  - Requires DHCP snooping in order to function
  - If a client tries to send traffic without a DHCP received address, it will be dropped
  - Clients using static ip-addresses need to be specifically allowed with custom entries (see below)
- IP source guard is a port-based feature that automatically creates an implicit (dynamic) port access control list (PACL)
  - Only supported in the ingress direction on L2 ports (cannot configure source guard on routed port)
  - Not supported on private-VLANs
- Optionally extend the functionality by also verifying the source mac-address with the port-security keyword
  - In this case the source mac-address must be identical to the learned address (in the CAM and DHCP snooping table)

```
interface
 ip verify source port-security
```

**ip verify source** - Enable ip source guard with source IP filtering

**ip verify source port-security** - Enable ip source guard with source IP and MAC address filtering

- Has no effect unless you also configure port-security on the interface
- If you enable this source-guard alongside port-security, the DHCP server must support the DHCP information option 82

```
interface
 access-group mode merge | prefer port
```

**prefer port** - IP source guard overrides other VACL configurations
**merge** - IP source guard and VACL configurations are both implemented and merged (default mode)

```
ip dhcp snooping
ip dhcp snooping vlan 10

interface gi1/0/2
 switchport access vlan 10
 ip verify source

interface gi1/0/3
 switchport access vlan 10
 ip verify source port-security
```

```
debug ip verify source packet
```

```
show ip verify source
Interface  Filter-type  Filter-mode  IP-address       Mac-address        Vlan
---------  -----------  -----------  ---------------  -----------------  ----
Gi1/0/2      ip           active       10.0.0.11                         10
Gi1/0/3      ip-mac       active       10.0.0.12        permit-all       10
```

## IP Source Guard Custom Bindings (Static IP)

- Use to allow traffic from hosts (servers) with static ip-addresses

**ip source binding** *mac-address* **vlan** *vlan-id ip-address* **interface** *interface-name*

```
ip source binding 0000.0000.0001 vlan 10 10.0.0.101 interface gi1/0/1
```

```
show ip source binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interfac
------------------  ---------------  ----------  -------------  ----  --------
00:00:00:00:00:01   10.0.0.101       infinite    static         10    GigabitE
00:02:B3:3F:3B:99   10.0.0.11        6522        dhcp-snooping  10    GigabitE
00:03:43:1B:00:9C   10.0.0.12        6522        dhcp-snooping  10    GigabitE
```

# IGMP Snooping

## L2 Multicast Address Conversion

- 1st byte of IP address is irrelevant, replace with 01-00-5E
- Convert the 2nd byte to binary and set the first bit to 0
- Convert the 2nd, 3rd and 4th byte to hex

In this case the 230.255.124.2 and 227.127.124.2 lead to exactly the same L2 address

- A 2nd byte of 255 or 127 will always result in the same value

| IPv4 address | 1st byte | 2nd byte | 3rd byte | 4th byte | L2 address |
|---|---|---|---|---|---|
| 226.144.154.4 | 01-00-5E | 00100000 = 10 | 9A | 04 | 01-00-5E-10-9A-04 |
| 230.255.124.2 | 01-00-5E | 01111111 = 7F | 7C | 02 | 01-00-5E-7F-7C-02 |
| 227.127.124.2 | 01-00-5E | 01111111 = 7F | 7C | 02 | 01-00-5E-7F-7C-02 |

- The mac-address consists of 48bits = 6 bytes
- The first 3 bytes (24 bits) are ways 01-00-5E
- The last 3 bytes are converted from the IP address
  - The first bit of the last 3 bytes is always 0
  - The last 23 bits are converted normally using method above

## IGMP Snooping / Querier

- The switch examines IGMP messages and learns the location of mrouters and hosts
- Listens for IGMP Reports/Leaves and limits multicast traffic to specific ports only
- Enable IGMP snooping globally or per VLAN (enabled by default on all VLANs)
- Can also bind static groups (using L2 address conversion) to interfaces

```
ip igmp snooping
ip igmp snooping vlan 10
ip igmp snooping vlan 1 static 01-00-5E-7F-7C-02 interface fa0/0

show mac address-table multicast
```

## IGMP Profiles

- IGMP Profile allows IGMP access-control at Layer 2
- Can only be applied to L2 interfaces
- Permit mode. Allows specified groups and blocks all others
- Deny mode. Blocks specified groups and allows all other

Only allow the specific multicast range

```
ip igmp profile 1
 permit
 range 224.0.O.O 229.255.255.255

int fa0/0
 ip igmp filter 1

show ip igmp profile
```

## PIM Snooping

- IGMP Snooping only limits traffic to and from hosts, PIM Snooping also limits traffic to mrouters
- Limits multicast traffic to interfaces that have downstream receivers joined to the same multicast group
- Listens to PIM hello, join, forward-election and prune messages
- Requires IGMP snooping and is applied on VLAN SVIs

```
interface vlan 10
 ip pim snooping
```

# IPv6 Security

# Destination Guard

## Destination Guard

- Destination Guard is a "last hop" security feature, the last hop router is the only one that is heavily impacted
- Interim routers don't have to NS for the final destination, they just CEF-switch the packet
- Needed because of the size of /64 IPv6 subnets and the possible amount of destinations

```
ipv6 destination-guard policy DESTINATION_POLICY
 enforcement always | stressed

vlan configuration 1
 ipv6 destination-guard attach-policy DESTINATION_POLICY

show ipv6 destination-guard policy
```

The stressed option will only enable Destination Guard during high usage

- Default is always

# DHCPv6 Guard

## DHCPv6 Guard

Trust all ports but require matching on link-local source and address range reply

```
ipv6 prefix-list TRUSTED_PREFIX permit 2001:10:0:12::/64 le 128
ipv6 access-list TRUSTED_SERVER
 permit ipv6 host FE80::1 any

ipv6 dhcp guard policy DHCP
 device-role server
 match server access-list TRUSTED_SERVER
 match reply prefix-list TRUSTED_PREFIX

vlan configuration 1
 ipv6 dhcp guard attach-policy DHCP
```

Or

```
interface vlan 1
 ipv6 dhcp guard attach-policy DHCP
```

## ND Inspection

- Control plane feature only, it doesn't inspect actual data traffic and only looks at ND ICMP packets.
- Builds a table based on NS/NA messages. It then enforces the table.
- If there is a link local address on the network, the switch will send a NS from the IPv6 address.
- If there isn't an IPv6 address on the VLAN (L2 switching only), it will send an NS from the IPv6 unspecified address.

- ND Tracking Policy is optional when enabling ND Inspection.
- Tracking is basically IP SLA echo only with ND packets. And is useful for two reasons:
    - Since the table is first-come first-serve, this frees up address space if it's actually not in use.
    - It allows for a host to move ports by aging out information.

```
vlan configuration 1
 ipv6 nd inspection
show ipv6 neighbor binding
```

Create a static binding to allow a certain host

```
ipv6 neighbor binding vlan 1 FE80::1 interface gi1 abcd.abcd.abcd

show ipv6 snooping capture-policy interface
```

Enable ND Inspection alongside a tracking policy

```
ipv6 nd inspection policy ND_POLICY
 tracking enable [reachable-lifetime] 300

vlan configuration 1
 ipv6 nd inspection attach-policy ND_POLICY
```

# IPv6 Snooping

## IPv6 Snooping

- Builds the neighbor database, similar to IPv6 ND inspection
- Glean ports are trusted ports
- The difference is that it uses and enforces more methods all at once
- It can use:
    - Information from DHCP (Default)
    - Information from ND (Default)
    - Static bindings

```
ipv6 snooping policy UNTRUSTED_HOSTS
 security-level guard | inspect

vlan configuration 1
 ipv6 snooping attach-policy UNTRUSTED_HOSTS
```

The **guard** keyword enables DHCP Guard, RA Guard, and ND Inspection.

The **inspect** keyword only enforces ND Inspection.

The policy is optional when enabling IPv6 Snooping on untrusted ports.

- By default, IPv6 snooping enables its version of RA Guard, DHCPv6 Guard and ND Inspection.
- Optionally disable ND Inspection with the no protocol ndp command.
- Optionally disable DHCPv6 Guard with the no protocol dhcp policy command.

```
ipv6 snooping policy TRUSTED_ROUTER
 security-level glean

int gi1
 ipv6 snooping attach-policy TRUST_ROUTER

show ipv6 neighbor binding
```

### IPv6 Source Guard

Any traffic other than IPv6 ND/RA and DHCP that doesn't match the source address present in the prebuilt binding table, will get dropped.

```
vlan configuration 1
 ipv6 source-guard
```

# RA Guard

### Router Advertisements (RA) Guard

- Apply on ports that should not receive router advertisements (RAs)
- In other words, another router should not be present on the link
- Configure on the switch to allow only RA from a single router on the specified port

```
ipv6 prefix-list RA_PREFIX permit 2001:10:0:12::/64
ipv6 access-list RA_SOURCE
 permit ipv6 host FE80::1 any

ipv6 nd raguard policy ROUTER
 device-role router
 match ra prefix-list RA_PREFIX (optional)
 match ipv6 access-list RA_SOURCE (optional)

int fa0/0
```

```
 description TRUSTED_ROUTER_PORT
 ipv6 nd raguard attach-policy ROUTER
```

Block all other RAs from other sources on the specified VLAN

```
ipv6 nd raguard policy HOSTS
 device-role host

ipv6 snooping logging packet drop
vlan configuration 1
 ipv6 nd raguard attach-policy HOSTS
```

Or

```
interface vlan 1
 ipv6 nd raguard attach-policy HOSTS
```

There is no need to specify a policy when applying RA Guard to untrusted hosts

- The **ipv6 nd raguard** command will suffice

The **ipv6 snooping logging packet drop** command is needed for logging untrusted RA messages

## RA Guard PACL

- It is also possible to configure the concept of RA Guard using a PACL to block unwanted RAs
- The **undetermintedtransport** keyword must be included to capture all unwanted traffic

```
ipv6 access-list RA_PACL
 deny icmp any any routeradvertisement
 deny ipv6 any any undeterminedtransport
 permit ipv6 any any

int fa0/0
 description UNTRUSTED_PORT
 ipv6 traffic-filter RA_PACL in
```

# PACL

## PACL / VACL Order of Preference (Interaction)

| Direction | Order |
|---|---|

| | |
|---|---|
| Ingress Direction | PACL is applied first<br>If packet is permitted by PACL, the VACL is consulted<br>If packet is permitted by VACL, access-list on L3 VLAN interface is consulted |
| Egress Direction | Access-list on L3 VLAN interface is applied first<br>If packet is permitted by L3 VLAN access-list, the VACL is consulted<br>No support for PACL in egress direction |

## Port-Based Access Control Lists (PACL)

- Can only be applied in the ingress direction
- Only supports one IPv4 access-list and one mac access-list per interface (two total)
- Does not support IPv6, MPLS or ARP traffic
- Does not support the access-list **log** and **reflect/evaluate** (statements are configurable, but ignored in IOS)

```
interface
 access-group mode merge | prefer port
```

**prefer port** - Overrides the effect of L3 VLAN interface access-lists or VACLs

- Exception is when traffic is sent to the route-processor (rp) due to **log** keyword for example
    - In this case the other filtering methods are consulted after the PACL

**merge** - PACL, VACL, and ACLs are merged in the ingress direction (default mode)

```
interface
 mac access-group acl-name in
 ip access-group acl-name in
```

```
ip access-list extended ICMP_HOST1
 permit icmp host 10.0.0.11 host 10.0.0.12

mac access-list extended MAC_HOST1
 permit host 0050.7966.6800 host 0050.7966.6801

interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 ip access-group ICMP_HOST1 in
 mac access-group MAC_HOST1 in
```

```
interface vlan 10
 ip add 10.0.0.1 255.255.255.0
 no shutdown
```

```
show access-list
Extended IP access list ICMP_HOST1
    10 permit icmp host 10.0.0.11 host 10.0.0.12
    20 deny ip any any
Extended MAC access list MAC_HOST1
    permit host 0050.7966.6800 host 0050.7966.6801
    deny   any any
```

```
show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  10    0050.7966.6800    DYNAMIC   Gi1/0/1
  10    0050.7966.6801    DYNAMIC   Gi1/0/2
```

# Port Security

## Port-Security

- Control port access based on host source mac-addresses
- Prevents users from plugging in (unmanaged) switches
    - Define how many mac-addresses can be learned on a specific port
    - Optionally statically define the source mac-addresses that are allowed to connect
- The port does not forward packets that do not conform
    - Can also be configured to err-disable ports or send SNMP traps when violations occur

## Port-Security Actions (Violation Modes)

| Mode | Port Action | Administrative Action |
|---|---|---|
| Protect | Traffic from violating hosts is dropped<br>No record of the violation is kept | Port will automatically resume after violations stop |

| | | |
|---|---|---|
| Restrict | Traffic from violating hosts is dropped<br>Record of the violation is kept<br>Switch can send a SNMP trap and/or syslog message | Port will automatically resume after violations stop |
| Shutdown (default) | Port is error-disabled (shutdown)<br>Optionally error-disable the entire VLAN with the **vlan** keyword | Port must be re-enabled manually (shut / no shut)<br>Port will automatically resume if **err-disable recovery** is configured |

**switchport port-security violation** *shutdown* | *restrict* | *protect*

Default violation mode is shutdown

- Enable automatic err-disable recovery with the **errdisable recovery cause psecure-violation** global command

## Port-Security Configuration

- Can only be configured on static access ports or trunk ports, not dynamic (DTP) ports
- Not supported on Private-VLAN ports
- When using the voice VLAN, set the maximum number of MAC-addresses for both the access and voice VLAN

```
interface
 switchport port-security
```

This turns on port-security on the interface (required), also adds:

```
switchport port-security violation shutdown
switchport port-security aging type absolute
switchport port-security aging time 0
switchport port-security maximum 1
```

- If you try to configure this on a DTP port, you will get the error message *Command rejected: Ethernet0/1 is a dynamic port*
- Default is aging absolute with a time of 0, this means the secure-mac-address never times out

```
interface gi1/0/1
 switchport mode access
```

```
switchport access vlan 10
switchport port-security
```

```
show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  10    0050.7966.6800    STATIC      Gi1/0/1
```

The dynamic mac-address in the CAM table will turn into a static address and will never time out

```
show port-security interface gi1/0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0050.7966.6800:10
Security Violation Count   : 0
```

Interfaces dynamically learn up to the maximum number of addresses allowed

- Unless you hardcode the allowed address, the first host mac-address will be learned
- Learned addresses do not age out by default (unless switch is rebooted)
    - Addresses are stored in the running-config and the CAM table
    - Make the learned addresses persistent with the **sticky** keyword

**switchport port-security maximum** *1-1024*

**maximum** - Specify the maximum mac-addresses that can be learned on the port (default is 1)

**switchport port-security mac-address** *mac-entry* | *sticky*

## Secure MAC-Address Types

| Type | Cisco Definition | Description | Times out by default | Times out with aging |
|---|---|---|---|---|
| Static | SecureConfigured | Manually defined and stored in running-config | no | yes (with **static** keyword) |
| Dynamic | SecureDynamic | Learned automatically until limit is reached<br>Removed when switch is rebooted | no | yes |
| Sticky | SecureSticky | Learned automatically until limit is reached<br>Stored in running/startup config (persistent) | no | no |

- Learned dynamic secure mac-addresses are automatically converted to sticky
  - If no mac-address is present on the interface, the first learned address will become sticky
- When using a statically defined address, the switch will still dynamically learn addresses if maximum value is more than 1

Only allow a single statically defined address

```
interface gi1/0/2
 switchport mode access
 switchport access vlan 10
 switchport port-security
 switchport port-security maximum 1
 switchport port-security mac-address 0050.7966.6801
```

```
show port-security address
              Secure Mac Address Table
------------------------------------------------------------------------
Vlan    Mac Address       Type                          Ports   Remaining Age
                                                                   (mins)
----    -----------       ----                          -----   -------------
  10    0050.7966.6800    SecureDynamic                 Gi1/0/1         -
  10    0050.7966.6801    SecureConfigured              Gi1/0/2         -
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 4096
```

Allow five persistent dynamic addresses

```
interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 switchport port-security
 switchport port-security maximum 5
 switchport port-security mac-address sticky
```

```
show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)          (Count)
-----------------------------------------------------------------------------
    Gi1/0/1              5            1                  0         Shutdown
    Gi1/0/2              1            1                  0         Shutdown
-----------------------------------------------------------------------------
```

Detect violations

```
interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 switchport port-security
 switchport port-security maximum 1
 switchport port-security mac-address 0000.0000.0001
 switchport port-security violation restrict
```

```
show port-security interface gi1/0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Restrict
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0050.7966.6800:10
Security Violation Count   : 3
```

```
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred,
caused by MAC address 0050.7966.6800 on port GigabitEthernet1/0/1
```

Clear dynamically learned mac-addresses on a port (allow other host to use the port)

- This can be used when the protect/restrict condition is in effect

```
clear port-security all | configured | dynamic | sticky
```

## Port-Security Aging

- Static address aging is disabled by default
- Sticky addresses never time-out
- After address age out, they are also removed from the CAM table (mac-address table)

| Aging Type | Description | Default value | Applies to |
|---|---|---|---|
| Absolute | Addresses time-out after specified time (default) | 0 (never) | Static / Dynamic |
| Inactivity | Address only time out after being inactive for a certain period of time | disabled | Static / Dynamic |

- Enable static aging with the **switchport port-security aging static** command
- Dynamic addresses that are converted to **sticky** will lose aging limitations

```
switchport port-security aging time 1-1440 (minutes)
switchport port-security aging type absolute | inactivity
switchport port-security aging static
```

- Inactivity timer is reset each time a frame is received from the host

Dynamic port-security with absolute aging

```
interface gi1/0/1
 switchport access vlan 10
 switchport mode access
 switchport port-security
 switchport port-security aging type absolute
 switchport port-security aging time 1
```

Static port-security with inactivity aging

```
interface gi1/0/2
 switchport access vlan 10
 switchport mode access
 switchport port-security mac-address 0050.7966.6801
 switchport port-security aging type inactivity
```

```
switchport port-security aging static
switchport port-security aging time 1
```

```
show port-security address
              Secure Mac Address Table
-----------------------------------------------------------------------------
Vlan    Mac Address       Type                          Ports   Remaining Age
                                                                    (mins)
----    -----------       ----                          -----   -------------
  10    0050.7966.6800    SecureDynamic                 Gi1/0/1      < 1
  10    0050.7966.6801    SecureConfigured              Gi1/0/2        1 (I)
-----------------------------------------------------------------------------
```

# Protected Ports

## Protected Ports

- Does not forward any traffic to any other port that is also a protected port
  - Same concept as isolated Private-VLAN ports
- Only control traffic (PIM for example) is forwarded
- All data traffic passing between protected ports must be forwarded through a Layer 3 device

```
interface fa0/0
 switchport protected
```

# Storm Control

## Storm Control

- Storm control monitors incoming traffic on the interface and applies limitations on flooding
- Limitation is applied before the traffic is flooded
- Storm control can apply to these traffic types (see MAC / (T)CAM / SDM section)
  - Broadcast frames
  - Multicast frames
  - Unknown unicast frames

- Monitors traffic in 1-second intervals
- The current traffic level of the interface is compared to the configured value every second
  - The level can be percentage of the total available (configured) bandwidth of the port

    - You can also a hardcoded the traffic level in total bps (bits per second) or pps (packets per second)
- The values can be different for each traffic type listed above

Storm control can be applied to a port-channel and all bundled interface

- However, you cannot apply storm control to a single interface bundled in the etherchannel
- If you try to apply storm control to a single bundled interface, the interface will be suspended from the etherchannel (s)

```
interface
 storm-control broadcast | multicast | unicast level rising % falling %
```

- Configures storm control based on bandwidth percentage (0-100)
    - 100 percent means no traffic storm control
    - 0 percent suppresses all traffic for that type

```
interface
 storm-control broadcast | multicast | unicast level bps rising falling
```

- Configures storm control based on bits per second

```
interface
 storm-control broadcast | multicast | unicast level pps rising falling
```

- Configures storm control based on packets per second

## Storm Control Thresholds

| Threshold | Value # | Description | Default |
|---|---|---|---|
| Rising | 1 | At this threshold, storm control starts dropping traffic | yes |
| Falling | 2 | At this threshold, storm control stops dropping traffic | Automatically set to the rising value |

- The normal transmission restarts when traffic drops below the falling threshold
    - If you don't specify a falling threshold, the rising value will be copied
    - This means that as soon as the traffic drops below the rising value, it is allowed again
- The benefit of the falling threshold is to drop traffic until it is lower than a certain value
    - This prevents the storm control from triggering on/off, on/off, etc.
- The falling value only really means something when the trap action is used
    - Otherwise the port is simply shutdown when the rising value is triggered

## Storm Control Actions

| Mode | Port Action | Administrative Action |
|---|---|---|
| Trap (default) | Traffic from violating hosts is dropped<br>Switch can send a SNMP trap message | Port will automatically resume after violations stop |
| Shutdown | Port is error-disabled (shutdown) | Port must be re-enabled manually (shut / no shut)<br>Port will automatically resume if **err-disable recovery** is configured |

**storm-control action** *shutdown* | *trap*

**snmp-server enable traps** *storm-control*
**errdisable recovery cause** *storm-control*

## Storm Control Configuration

Percentage based

```
interface gi1/01
 bandwidth 1000000
 storm-control broadcast level 50 25
 storm-control action trap
```

- Storm control will start dropping traffic after broadcast traffic consumes 50% of the link
- When the broadcast storm drops under 25% the port stops dropping traffic

```
show storm-control gi1/0/1 broadcast
Interface  Filter State   Upper     Lower     Current
---------  -------------  --------  --------  --------
Gi1/0/1    Forwarding     50.00%    25.00%    0.00%
```

Packets per second based

```
interface gi1/01
 storm-control broadcast level pps 500
 storm-control action trap
```

```
show storm-control gi1/0/1 broadcast
Interface  Filter State   Upper     Lower     Current
---------  -------------  -------   -------   -------
Gi1/0/1    Blocking       500 pps   500 pps   800 pps
```

- Storm control will start dropping traffic after broadcast traffic reaches 500 pps
- When the broadcast storm drops under 500 pps the port stops dropping traffic
    - Notice that the falling threshold is automatically set to 500

Bits per second based

```
interface gi1/01
 storm-control unicast level bps 1000
 storm-control action shutdown

errdisable recovery cause storm-control
errdisable recovery interval 60
```

```
show storm-control gi1/0/1 unicast
Interface  Filter State   Upper     Lower     Current
---------  -------------  --------  --------  -------
Gi1/0/1    Forwarding     1000 bps  1000 bps  300 bps
```

- Storm control will shut down the port after unknown unicast traffic reaches 1000 bps
- After 60 seconds the port is re-enabled

## Small-Frame Storm Control

- Incoming VLAN-tagged frames smaller than 67 bytes are considered small frames
- Do not cause the switch storm-control counters to increment
- Can only be configured for pps, not level or bps

```
interface
 small violation rate pps
```

```
interface gi1/0/1
 small violation-rate 1000
```

# VACL

## PACL / VACL Order of Preference (Interaction)

| Direction | Order |
|---|---|
| Ingress Direction | PACL is applied first<br>If packet is permitted by PACL, the VACL is consulted<br>If packet is permitted by VACL, access-list on L3 VLAN interface is consulted |
| Egress Direction | Access-list on L3 VLAN interface is applied first<br>If packet is permitted by L3 VLAN access-list, the VACL is consulted<br>No support for PACL in egress direction |

## VLAN Access Control Lists (VACL)

- Applies to VLANs and can be applied to any (present) VLAN
  - Can filter inter-VLAN traffic (between hosts in different VLANs) or between hosts in the same VLAN
  - Can also intercept traffic and forward to another VLAN, similar to PBR and is called Policy-Based Forwarding (PBF)
  - Cannot filter multicast traffic such as PIM, IGMP and MLD
- Can filter both L2 frames (mac access-lists) and L3 packets (ip access-lists)
- Uses vlan access-maps, which work very similarly to route-maps (sequence numbers)
  - Define an action (drop / forward / redirect) in the sequence while matching on an ACL
- Like route-maps there is an implicit deny at the end of the vlan access-map
  - This implicit deny will only apply to the type of traffic matched in the access-list (L2 or L3)
  - For example a vlan access-map with an implicit deny at the end for L3 traffic will still allow L2 traffic
- You can match multiple ACLs in a single vlan access-map sequence statement

**vlan access-map** *map_name 0 -65535 (default sequence is 10, 20, etc.)*
**match ip address** *ip_acl_name* **| mac address** *mac_acl_name*
**action** *drop* | *forward* | *redirect*

**drop** - Drop the packet/frame matched in the access-list

- Add the **log** keyword to create a syslog entry for dropped traffic (only dropped traffic can be logged)

**forward** - Forward the packet/frame to the original destination

- Packets can still be dropped on another access-list applied to the L3 VLAN interface
- Add the **vlan** keyword change the destination and to forward the traffic to another VLAN (PBF)
  - See Policy-Based Forwarding (PBF) section below
  - Only 1 VLAN is supported in the **vlan filter** command

    - Use the **local** keyword to allow PBF hosts to communicate in the same VLAN (not possible by default)
- Add the **capture** keyword to duplicate the traffic to another port configured for capturing (see VLAN Capture Port below)
    - Traffic is still forwarded to the original destination (only forwarded traffic can be captured)
    - This functionality is basically a VLAN variant of SPAN

**redirect** - Change the destination and to forward the traffic to another interface (up to 5)

- The redirect interface must be in the VLAN for which the vlan access-map is configured
- Can be used to intercept traffic and forward it to a SPAN (destination) port
    - Cannot be used to redirect traffic to a L3 VLAN interface

**vlan filter** *map_name* **vlan-list** *vlan_list* | *interface*

- VLAN access-maps can be applied to multiple VLANs (range or comma-separated)
- Per VLAN you can only use a single vlan access-map

Only allow hosts with source ip-address 10.0.0.0/24

```
ip access-list extended IP_VLAN10
 permit ip 10.0.0.0 0.0.0.255 any

vlan access-map VACL_VLAN10
 match ip address IP_VLAN10
 action forward

vlan filter VACL_VLAN10 vlan-list 10
```

```
show vlan access-map
 Vlan access-map VACL_VLAN10 10
 match: ip address IP_VLAN10
 action: forward
```

Only allow hosts which have NICs from certain vendors (0050.79)

```
mac access-list extended OUI
 permit 0050.7900.0000 0000.00ff.ffff any

vlan access-map VACL_IOU
 match mac address MAC
 action forward
```

```
vlan filter VACL_IOU vlan-list 1-4094
```

```
show access-lists
Extended MAC access list OUI
    permit 0050.7900.0000 0000.00ff.ffff any
```

```
show vlan filter
VLAN Map VACL_IOU:
        Configured on VLANs:  1-4094
            Active on VLANs:  10
```

## VLAN Capture Port

- Any port can function as a capture port
  - A capture port provides basically the same functionality as a SPAN destination port
- Multiple VLANs can be redirected to a single capture port

Capture all traffic from host1, regardless of which VLAN it is connected to (10 or 20)

```
mac access-list extended CAPTURE_HOST1
 permit host 0050.7966.6800 any

vlan access-map VACL_HOST1
 match mac address CAPTURE_HOST1
 action forward capture
vlan access-map VACL_HOST1 99
 action forward
vlan filter VACL_HOST1 vlan-list 10,20

interface gi1/0/1
 switchport capture
```

- If you don't specify the second sequence statement (99) all other traffic will be dropped

```
show vlan filter
VLAN Map VACL_HOST1:
        Configured on VLANs:  1-4094
            Active on VLANs:  10
```

## VLAN Policy-Based Forwarding (PBF) / MAC PBF

- Remember bi-directional traffic flows

    - To allow PBF traffic in both directions between two VLANs, you need to apply the rule to both VLANs
- Can be configured between hosts that are connected to different switches
- By default, MAC PBF hosts in the same VLAN cannot communicate with each other (because it forwards to another VLAN)
    - Override this behavior with the **local** keyword on the **forward** vlan access-map statement
- VLAN access-maps using PBF can only be applied to a single VLAN
    - If applied to multiple VLANs (using the vlan filter command) it is only applied to the last VLAN in the list
    - The command **vlan filter VACL_PBF vlan-list 10-100,200** will only apply the vlan access-map to VLAN 200
- PBF is a L2 function and does not need an operational L3 interface VLAN
- In a way, you can use PBF to provide inter-VLAN 'routing' without a L3 interface

```
interface gi1/0/1
 description HOST1
 switchport access vlan 10
interface gi1/0/2
 description HOST2
 switchport access vlan 20

mac access-list extended HOST1
  permit host 0050.7966.6800 host 0050.7966.6801

mac access-list extended HOST2
  permit host 0050.7966.6801 host 0050.7966.6800

vlan access-map VACL_HOST1
 match mac address HOST1
 action forward vlan 20
vlan access-map VACL_HOST1 99
 action forward

vlan access-map VACL_HOST2
 match mac address HOST2
 action forward vlan 10
vlan access-map VACL_HOST2 99
 action forward

vlan filter VACL_HOST1 vlan-list 10
vlan filter VACL_HOST2 vlan-list 20
```

- Host1 is in VLAN 10 on gi1/0/1 using mac-address 0050.7966.6800
- Host2 is in VLAN 20 on gi1/0/2 using mac-address 0050.7966.6801
- MAC PBF will act as a relay between VLAN 10 and 20 for Host1 and 2
    - All other traffic is forwarded normally

## VACL Logging

- Only traffic that is dropped can be logged
    - Only denied IP packets are logged
    - Logs are generated based on flows, meaning that not every single packet is logged
    - Log is generated when first matching packet in the flow is received
    - Alternatively you can configure a threshold which will only start logging at a certain amount of packets
- Logs are stored in the VLAN access-log table
    - This table is finite and holds 500 flows by default
    - When the log table is full, new logs entries are not written to the table
    - The contents of the table can be cleared by setting the **maxflow** value to 0
- Instead of logging every single flow, you can configure a log threshold
    - This threshold defines when logging of the flow should start (after certain amount of packets)
    - Disabled by default
- You can also rate-limit the amount of packets for per second
    - Enabled by default at 2000 packets per second (pps)

**vlan access-log maxflow** *max_number (0-2048*)

- The amount of flows that are stored in the table (500 flows by default)

**vlan access-log ratelimit** *pps (0-5000)*

- The maximum redirect VACL logging packet rate, exceeding packets are dropped (default is 2000 pps)

**vlan access-log threshold** *packet_count*

```
vlan access-log maxflow 500
vlan access-log ratelimit 2000
vlan access-log threshold 1000
```

```
show vlan access-log config

VACL Logging Configuration:
        max log table size      :500
        log threshold           :1000
        rate limiter            :2000
```

```
show vlan access-log

id  prot    src_ip          dst_ip      sport  dport vlan    port
lastlog

 ------------------------------------------------------------------------
 1  17      10.0.0.11       10.0.0.12    68     67     10    Gi1/0/1
 2  17      10.0.0.11       10.0.0.12    68     67     10    Gi1/0/1
 3  17      10.0.0.11       10.0.0.12    68     67     10    Gi1/0/1
 4  17      10.0.0.11       10.0.0.12    68     67     10    Gi1/0/1
 5  17      10.0.0.11       10.0.0.12    68     67     10    Gi1/0/1

 Total number of matched entries: 5
```

### VACL for Private-VLANs

- Host port to promiscuous port traffic is matched on secondary private-VLAN access-list
- Promiscuous port to host port traffic is matched on primary private-VLAN access-list
- To filter out specific IP traffic apply the VACL to both the primary and secondary private-VLANs

# VLANs & Trunking

### Built-in VLANs

| VLAN ID | Description |
|---|---|
| VLAN 1 | default |
| VLAN 1002 | fddi-default |
| VLAN 1003 | token-ring-default |
| VLAN 1004 | fddi-net-default |
| VLAN 1005 | token-ring-net-default |

## VLAN Configuration

- Default VLAN name when creating a VLAN is VLANXXXX, where XXXX represents the VLAN number with leading zeroes
    - For example VLAN 100 is named VLAN0100 by default
- Every VLAN except VLAN1 can be renamed, the name for VLAN1 is always default

```
vlan 100
vlan 200
vlan 300

int gi1/0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
int gi1/0/1
 switchport mode access
 switchport access vlan 100
int gi1/0/2
 switchport mode access
 switchport access vlan 200
int gi1/0/3
 switchport mode access
 switchport access vlan 300
```

```
show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- ---------
1    default                          active
100  VLAN0100                         active    Gi1/0/1
200  VLAN0200                         active    Gi1/0/2
300  VLAN0300                         active    Gi1/0/3
1002 fddi-default                     act/unsup
1003 trcrf-default                    act/unsup
1004 fddinet-default                  act/unsup
1005 trbrf-default                    act/unsup
```

```
show interfaces trunk
Port        Mode             Encapsulation  Status        Native vlan
Gi1/0/1     on               802.1q         trunking      1
Port        Vlans allowed on trunk
Gi1/0/1     1-4094
Port        Vlans allowed and active in management domain
Gi1/0/1     1,100,200,300
Port        Vlans in spanning tree forwarding state and not pruned
Gi1/0/1     1,100,200,300
```

- If you remove a VLAN that has active ports, the ports will go into an inactive state until they are added to a new VLAN

```
vlan 10
interface gi1/0/1
 switchport mode access vlan 10

no vlan 10
```

```
show interface gi1/0/1 switchport

Name: Gi1/0/1
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: static access
Access Mode VLAN: 10 (Inactive)
```

## VLAN Database

- When the switch is in VTP server or transparent mode, you can configure VLANs in the VLAN database mode
- When you configure VLANs in VLAN database mode, the VLAN configuration is saved in the vlan.dat file

Extended Range VLANs & VTP

- Normal range VLANs (1-1001) are stored in the vlan.dat file
- Extended range VLANs (1006-4094) are not stored in the vlan.dat file
- You can only configure extended range VLANs in VTPv3 (all modes) or VTPv1/2 mode transparent
    - Extended range VLANs need to be remapped if the switch is put back from transparent into VTP server/client mode

## Routed Interface VLAN Allocation

- A routed interface on the switch is assigned a VLAN from 1006 and up by default

- These VLANs will become unusable for their normal purpose
- Can be altered to descend from VLAN 4094 downwards with the **vlan internal allocation policy** command (if supported)

**vlan internal allocation policy** *ascending* | *descending*

```
show run all | i internal

vlan internal allocation policy descending
```

```
interface gi1/0/1
 no switchport
 ip address 10.0.0.1 255.255.255.0
```

```
show vlan internal usage

VLAN Usage
---- --------------------
4094 GigabitEthernet1/0/1
```

## VLAN Trunking Protocols

| Protocol | Overhead | Source VLAN | Workings | Concept of native VLAN |
|---|---|---|---|---|
| ISL | 26 byte header<br>4 byte CRC trailer | 15-bit field<br>Range 1-4094 | Encapsulates entire frame and adds new header + trailer<br>Total overhead is 30 bytes, not effective<br>Destination mac-address is changed | no |
| 802.1Q | 4 byte tag | 12-bit field<br>Range 0-4095<br>0,1,4095 reserved | Embeds tag directly in the L2 frame<br>FCS (Frame Check Sequence) is re-calculated after insertion<br>Total overhead is 4 bytes, effective<br>Destination mac-address is not changed | yes |

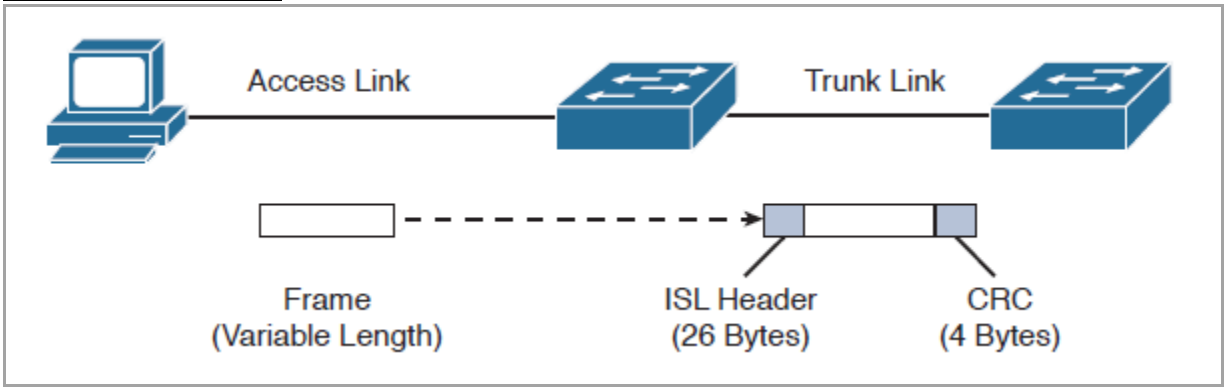
- 802.1Q is also referred to as 'simple or internal tagging'
- Introduces the concept of native VLANs (can be user-defined, default is 1)

- Frames in this VLAN do not receive a dot1q tag (can be overridden with the **vlan dot1q tag native** global command)
- The native VLAN is mostly used for end-devices that connect through trunk ports (usually through a phone)
- Any untagged traffic arriving on the trunk port (from the end-station) is placed in the native VLAN without a tag
- See Native VLAN section for more information

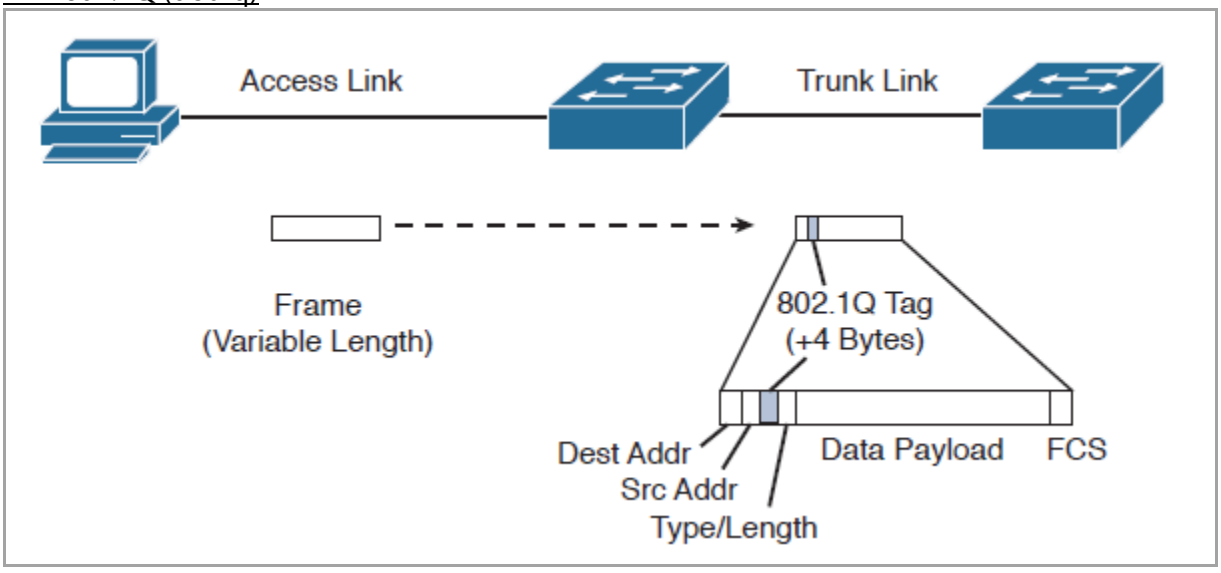
**vlan dot1q tag native** - A trunk will always perform tagging on the outgoing frames

- The native VLAN setting is ignored and all frames are tagged with the corresponding tag value
- Untagged frames arriving at a trunk port will be dropped without being forwarded further
- Essentially this command disables the regular workings of the native VLAN
- Has no effect on access-ports

Inter-Switch Link (ISL)




IEEE 802.1Q (dot1q)




Both ISL and dot1q add bytes to the overall frame (30 for ISL and 4 for dot1q)

- Ethernet frames cannot exceed 1518 bytes
- Trunking protocols can cause the frame to become too large and be dropped (errors on link)
    - Frames that barely exceed the MTU size are called baby giant frames
- ISL uses proprietary hardware with ISL to ensure frames are not oversized
- 802.1Q complies with 802.3ac, which increases the maximum frame size to 1522 (1518 + 4)
    - The MTU is 1500 bytes (with the header (18) and dot1q it is 1522)

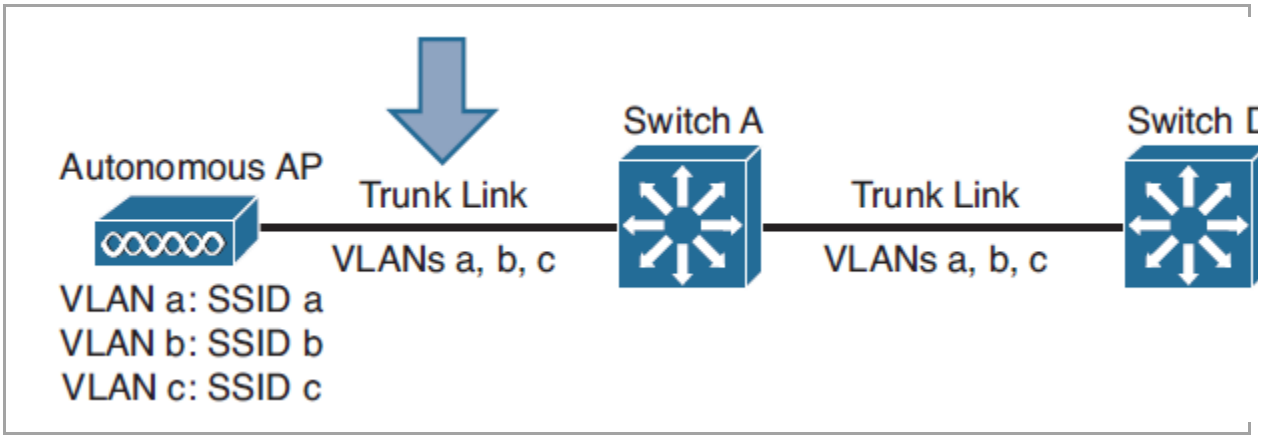
| |
|---|
| switchport mode **dynamic** *desirable* \| *auto*<br>switchport mode trunk |

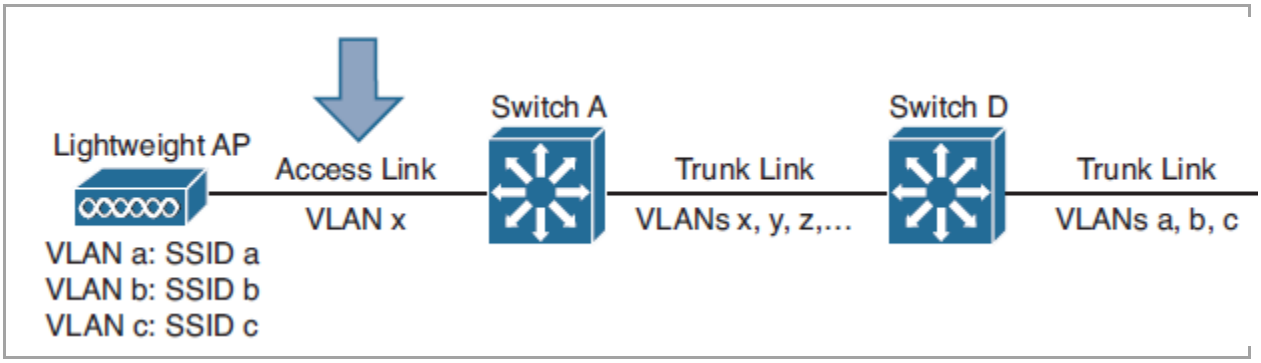
| |
|---|
| switchport trunk **encapsulation** *isl* \| *dot1q* \| *negotiate* |

## Wireless VLANs

Cisco APs can operate in one of the two following modes:

- Autonomous mode - The AP operates independently and directly connects VLANs to WLANs on a one-to-one basis
    - Generally connects to trunk links that carry multiple VLANs
- Lightweight mode - The AP must join and cooperate with a wireless LAN controller located elsewhere on the network
    - The AP connects each of its own WLANs with a VLAN connected to the controller
    - All of the VLAN-WLAN traffic is encapsulated and carried over a special tunnel between the AP and the controller
    - Generally connect to access ports that carry a single VLAN






# DTP

## Dynamic Trunking Protocol (DTP)

- DTP Frames negotiate the port state (trunk or access) and the trunking mode
  - DTP frames are sent to the L2 multicast address 0100.0ccc.cccc
- Carries VTP domain information and sent every 30 seconds
  - If VTP domain does not matches, DTP will not negotiate a trunk
  - In this case the port state will always be access

**switchport mode dynamic** *desirable* | *auto*
**switchport trunk encapsulation** *isl* | *dot1q* | *negotiate*

ISL is the preferred encapsulation mode when using **negotiate**

## DTP Modes

- Default mode depends on switch platform
- Desirable has priority over auto
  - If one side is set to auto, and one side is desirable the port will become a trunk

If you configure one side as static trunk, and the other as DTP auto/desirable a trunk will form

- The side that is statically configured has to also configure the trunk encapsulation to isl/dot1q
- Trunk encapsulation negotiate only works with DTP auto / desirable ports, not static trunks
- The side that is still set to use DTP can negotiate and adjust its encapsulation to static side

| DTP Mode | Prefers |
|---|---|
| Desirable | Trunk |
| Auto | Access |

```
interface gi1/0/1
 switchport mode dynamic desirable
 switchport trunk encapsulation negotiate
```

```
show interfaces gi1/0/1 switchport
Name: Gi1/0/1
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: isl
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Appliance trust: none
```

- The most important information is the 'operational mode'
  - The administrative mode is what is configured on the port
  - Negotiation of trunking means whether DTP is on or not (**switchport nonegotiate**)
  - Manually defining a trunking mode does <u>not</u> disable DTP

```
show interfaces trunk
Port        Mode             Encapsulation  Status        Native vlan
Gi1/0/1     desirable        n-isl          trunking      1
Port        Vlans allowed on trunk
Gi1/0/1     1-4094
Port        Vlans allowed and active in management domain
Gi1/0/1     1
Port        Vlans in spanning tree forwarding state and not pruned
Gi1/0/1     1
```

```
show dtp
Global DTP information
        Sending DTP Hello packets every 30 seconds
        Dynamic Trunk timeout is 300 seconds
        4 interfaces using DTP
```

```
show dtp interface gi1/0/1
DTP information for GigabitEthernet1/0/01
  TOS/TAS/TNS:                              TRUNK/DESIRABLE/TRUNK
  TOT/TAT/TNT:                              ISL/NEGOTIATE/ISL
  Neighbor address 1:                       AABBCC000200
  Neighbor address 2:                       000000000000
  Hello timer expiration (sec/state):       25/RUNNING
  Access timer expiration (sec/state):      295/RUNNING
  Negotiation timer expiration (sec/state): never/STOPPED
  Multidrop timer expiration (sec/state):   never/STOPPED
  FSM state:                                S6:TRUNK
  # times multi & trunk                     0
  Enabled:                                  yes
  In STP:                                   no

  Statistics
  ----------
  4 packets received (4 good)
  0 packets dropped
      0 nonegotiate, 0 bad version, 0 domain mismatches,
      0 bad TLVs, 0 bad TAS, 0 bad TAT, 0 bad TOT, 0 other
  13 packets output (13 good)
      7 native, 6 software encap isl, 0 isl hardware native
  0 output errors
  0 trunk timeouts
  2 link ups, last link up on Thu Dec 29 2016, 08:59:09
  2 link downs, last link down on Thu Dec 29 2016, 08:54:24
```

Turn off DTP with the following commands

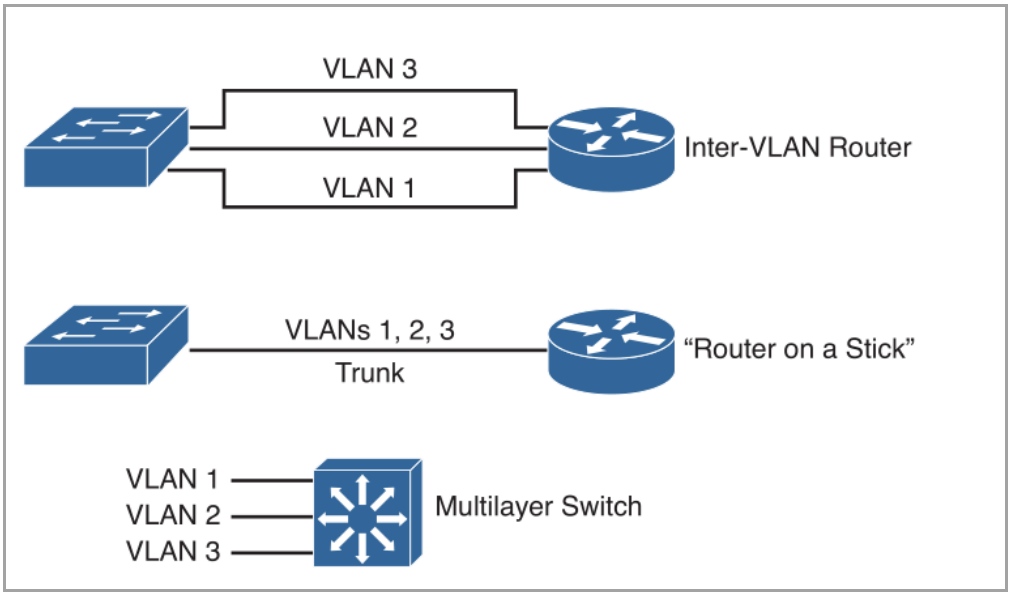
```
switchport nonegotiate
switchport mode access
no switchport
switchport mode private-vlan
switchport mode dot1q-tunnel
```

# Inter-VLAN / MLS

## Routing Between VLANs

Three options:

- Inter-VLAN Router - One router connected to multiple access interfaces in different VLANs
- Router on a Stick - One router with sub-interfaces connected to a trunk
- Multilayer Switch - Switch with routing capabilities




## Switched Virtual Interface (SVI)

- VLAN interfaces (usually) start in the administratively disabled state, always verify
  - The SVI will only be active if there are active access ports for the specific VLAN (or trunks)
  - The SVI will remain down, this is called the SVI autostate
  - Exclude ports from affecting the state of SVIs with **switchport autostate exclude** interface command
- Creating an interface for a non-existing VLAN will not automatically create the VLAN

## Multilayer Switching (MLS)

- The difference between MLS and L3 switching is that MLS also includes layers above L3 (L4 segment for example)
- Route caching (NetFlow LAN switching or flow-based switching)
    - First generation requiring a route processor (RP) and a switch engine (SE)
    - The RP uses the first packet in a flow to determine the destination
    - The SE uses the information from the first packet to create a shortcut entry in the cache and forwards  subsequent packets
- Topology based
    - Second generation of MLS that uses CEF

### Packet Rewrite

- When using MLS, the switch behaves very much as a router does (mac-address rewriting)
- A L2 switch just blindly forwards frames (transparent bridging)
- The L3 switch does the following with an incoming packet from a host on VLAN10 to VLAN20 for example:
    - Destination mac-address is changed from the original value (L3 VLAN interface) to next-hop device mac-address
    - Source mac-address is changed from the original value (host) to the L3 VLAN interface
    - TTL is decremented by 1 and L2 and L3 checksums are recalculated

# Native VLAN

### Native VLAN

- The native VLAN is a concept introduced by dot1q
    - All arriving traffic that is untagged is placed in the native VLAN
    - ISL tags all traffic, including the native VLAN
    - You can tag the native VLAN with  by configuring **vlan dot1q tag native** globally
- Primarily used for management traffic and VoIP phones nowadays
- Can be used by an ip-phone connected to the switch
    - Voice traffic will arrive tagged and data traffic from PC will arrive untagged
- LLDP and CDP use the native VLAN
- 802.1D (STP) frames are sent in the native VLAN

Frames belonging to the native VLAN are not encapsulated with any tagging information at all

- This is very similar to an access port and is as if a trunk link is not being used
- If an end-station connects to a trunk port, they can only understand the native VLAN frames because they are untagged

**vlan dot1q tag native** - A trunk will always perform tagging on the outgoing frames

- The native VLAN setting is ignored and all frames are tagged with the corresponding tag value

- Untagged frames arriving at a trunk port will be dropped without being forwarded further
- Essentially this command disables the regular workings of the native VLAN
- Has no effect on access-ports

Native VLAN Mismatch

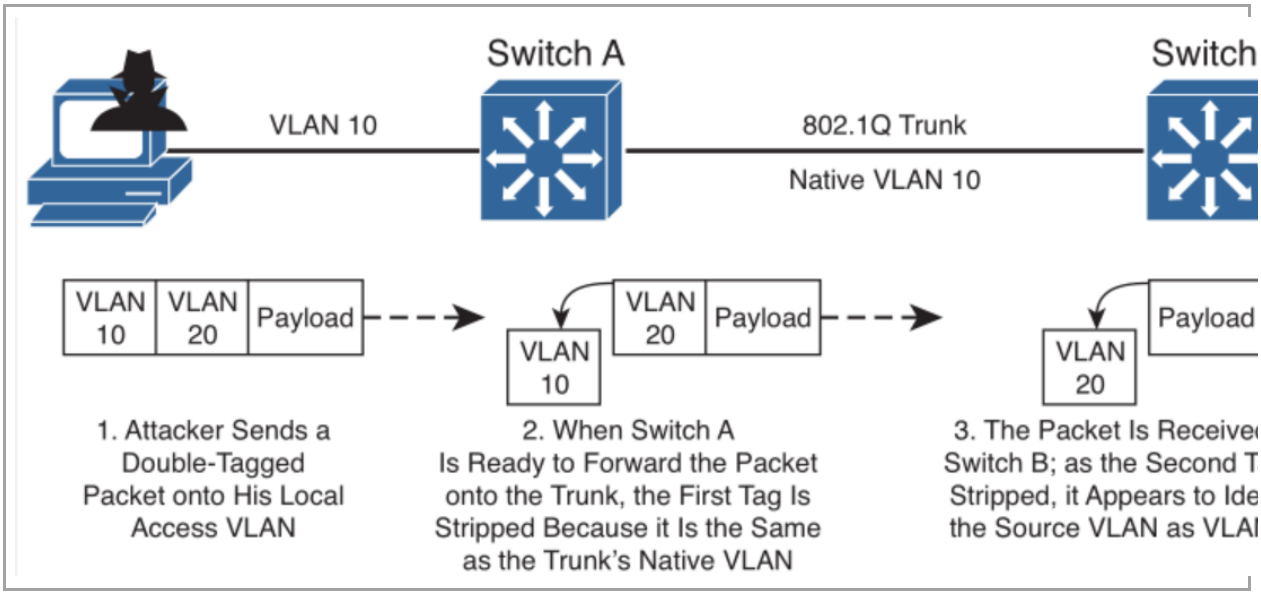
- The native VLAN is configured independently of the trunk encapsulation
  - It is possible to have a native VLAN mismatch even if the ports use ISL encapsulation
  - In this case, the mismatch is only cosmetic and will not cause a trunking problem
- Native VLAN mismatch syslog messages are sent from the CDP protocol (CDPv2 only)

Even though STP/CDP/DTP and PAgP (and others) use the native VLAN, you can set this VLAN to an bogus VLAN-id

- This action will prevent VLAN hopping (see below)
- The management protocols will not be affected if the native VLAN is removed or pruned from the link

## VLAN Hopping Attack

- An attacker connects to an access port (say VLAN10)
- The switch has VLAN10 as native on a trunk link to another switch
- The attacker sends frames with spoofed dot1q tags (double tags)
  - One tag (the first tag) is the actual VLAN it is connected to
  - The second tag is the spoofed VLAN that it tries to reach
- The switch strips the first tag and sends the frame over the trunk link
- The neighboring switch receives the frame with the spoofed tag and forwards it to the respective VLAN




- Negate this attack by setting the native VLAN to a bogus (unused) VLAN

    - Another method is to always tag the native VLAN with the **vlan dot1q tag native** global command

## Native VLAN Mismatch Trunk

Native VLAN mismatch syslog messages are sent from the CDP protocol (CDPv2 only)

- Traffic can actually flow between mismatched switches
- PVST will give errors with this configuration
    - Other STP (MSTP) configurations will work
- Native VLANs are useful when the same VLANs exist on opposite ends of the switches, but are not used for the same purpose
    - An example would be a company merger that uses the same VLAN-id's internally
    - For example: CSW2--->Native VLAN 2------Trunk Connection------Native VLAN 3<---CSW3

CSW2

```
interface fa0/1
 switchport access vlan 2

no spanning-tree vlan 2

interface fa0/24
 description CSW3
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 2
 switchport mode trunk
 no cdp enable
```

CSW3

```
interface fa0/1
 switchport access vlan 3

no spanning-tree vlan 3

interface fa0/24
 description CSW2
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 3
 switchport mode trunk
 no cdp enable
```

# Private VLANs

## Private VLANs

- All hosts in the private VLAN belong to the same broadcast domain / subnet as the primary PVLAN
- Private VLANs require VTPv3 or VTP mode transparent on older versions of VTP
- Private VLANs are not compatible with VTP pruning and requires that it is turned off (if using VTPv3)
- Enabling DHCP Snooping, ARP Inspection and Source Guard on primary PVLAN will also enable it on secondary PVLANs

| **Private VLAN** | **PVLAN Type** | **Description** | **Ports** | **Can talk to** | **Max #** |
|---|---|---|---|---|---|
| Primary | Primary | The broadcast domain with which the secondary VLANs are associated | Promiscuous | All | 1 |
| Community | Secondary | Can talk to members of the same community and the primary VLAN<br>Cannot talk to members of other communities<br>Cannot communicate with isolated ports | Community | Primary<br>Community | Multiple |
| Isolated | Secondary | Isolated ports can community only with the primary VLAN<br>Cannot communicate with community or other isolated ports | Isolated | Primary | 1 |

**vlan** *primary-vlan*
**private-vlan** primary

**vlan** *secondary-vlan*
**private-vlan** *community* | *isolated*

Associate secondary PVLANs with primary

**vlan** *primary-vlan*
**private-vlan association** *secondary-vlans*

## Private VLAN Ports

| PVLAN Port | Belongs to | Description |
|---|---|---|
| Promiscuous | Primary | Connects to L3 gateway device and is the exit interface for the secondary PVLANs |
| Host | Isolated | Connects to a host and is only allowed to communicate with the promiscuous port |
| Host | Community | Connects to a host and is only allowed to communicate with the promiscuous port and other ports in same community |

```
interface
 switchport mode private-vlan host | promiscuous
```

```
interface
 switchport private-vlan association host secondary_vlan
 switchport private-vlan association mapping primary_vlan secondary_vlans
```

**association host** - Links the host interface to a secondary PVLAN (old-style command)
**association mapping** - Links the promiscuous interface to a single primary, and multiple secondary PVLANs (old-style command)

```
interface
 switchport private-vlan host-association secondary_vlan
 switchport private-vlan mapping primary_vlan secondary_vlans
```

**host-association** - Links the host interface to a secondary PVLAN (new-style command)
**mapping** - Links the promiscuous interface to a single primary, and multiple secondary PVLANs (new-style command)

- The old style commands will end up in the running config as the new-style
- The new style commands try to differ between 'association' and mapping
    - This is why the documentation states that host ports are associated and promiscuous ports are mapped

## Private VLAN Trunks

- The ability to trunk private-VLANs between switches is dependent on the (PVLAN) capabilities of the switch
    - Between two switches that both support private-VLANs, there is no issue and you can use regular dot1q trunks

- Trunking to a switch that does not support private-VLANs needs a special PVLAN trunk port
    - These trunk ports basically merge the primary and secondary VLANs into a single VLAN (the primary)

| Interface Type | Used between switches | Functions as a | Switch (devices) on the other side can |
|---|---|---|---|
| Regular trunk port | That both support PVLANs | Regular trunk | Reach all secondary and primary PVLANs |
| Promiscuous PVLAN trunk | PVLAN switch and a non-supporting switch | promiscuous PVLAN port | Reach all secondary and primary PVLANs |
| Isolated PVLAN trunk | PVLAN switch and a non-supporting switch | isolated PVLAN port | Reach only the promiscuous port on the other switch Comparable to a local isolated PVLANport |

Both promiscuous and isolated PVLAN trunks merge the secondary and primary PVLAN into one (secondary is rewritten to match primary)

- These trunks connect to switches that do not support PVLANs
- The main difference between these ports is the PVLANs that the remote switch can reach
    - Isolated - The remote switch can 't reach any (community / isolated) PVLANs, and can only reach the promiscuous port (that probably connects to a L3 device)
    - Promiscuous - The remote switch can reach all (community / isolated) PVLANs

## Private-VLANs and VLAN interfaces (SVI)

- Switch is capable of inter-VLAN routing
    - Use when the switch is performing DHCP services or provides other L3 functionality
- Does not necessarily use a promiscuous port, instead the SVI will act as a promiscuous L3 interface
- You only need to configure the primary PVLAN L3 interface
    - Do not create any secondary PVLAN SVIs

```
interface vlan primary_vlan
 private-vlan mapping secondary_vlans
```

## Private VLANs Configuration

1. Set VTP mode / version
2. Create primary PVLAN and define
3. Create secondary PVLANs and define
4. Associate secondary to primary PVLAN

5. Associate hosts
6. Create promiscuous port, trunk or SVI

PVLAN with VTPv3 and promiscuous port

```
vtp version 3
no vtp pruning

vlan 100
  private-vlan primary
vlan 110
  private-vlan community
vlan 120
  private-vlan isolated
vlan 100
 private-vlan association 110,120

interface gi1/0/1
 description PROMISCUOUS
 switchport mode private-vlan promiscuous
 switchport private-vlan mapping 100 110,120

interface range gi1/0/2 - 3
 description COMMUNITY
 switchport mode private-vlan host
 switchport private-vlan host-association 100 110

interface gi1/0/4
 description ISOLATED
 switchport mode private-vlan host
 switchport private-vlan host-association 100 120
```

- Promiscuous ports show up as members of all secondary VLANs

```
show vlan private-vlan

Primary Secondary Type              Ports
------- --------- ----------------- -------------------------
100     110       community         Gi1/0/1, Gi1/0/2, Gi1/0/2
100     120       isolated          Gi1/0/1, Gi1/0/4
```

PVLAN with VTPv1/2 and SVI

```
vtp mode transparent
```

```
ip routing

vlan 100
  private-vlan primary
vlan 110
  private-vlan community
vlan 120
  private-vlan isolated
vlan 100
 private-vlan association 110,120

interface vlan 100
 ip address 10.0.0.1 255.255.255.0
 no shutdown
 description SVI
 private-vlan mapping 110,120

interface range gi1/0/2 - 3
 description COMMUNITY
 switchport mode private-vlan host
 switchport private-vlan host-association 100 110

interface gi1/0/4
 description ISOLATED
 switchport mode private-vlan host
 switchport private-vlan host-association 100 120
```

### Private VLAN Workarounds

- You can provide communication between isolated PVLAN ports using two methods:
  - Configure **ip local-proxy-arp** on the promiscuous port
  - Run OSPF between the clients and define the network as point-to-multipoint

IP-proxy-arp and local-proxy arp

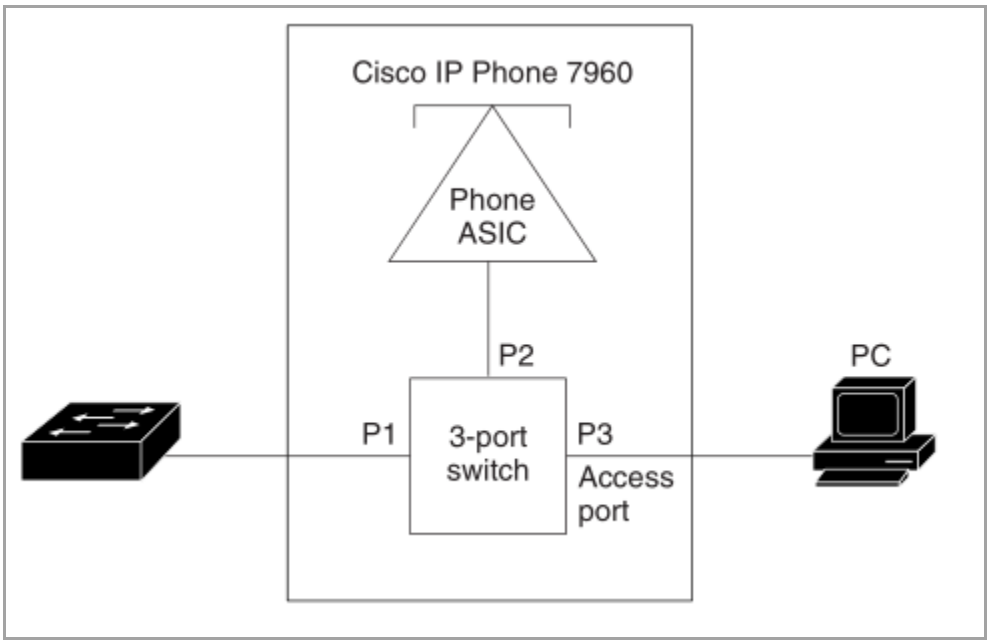
- The **ip proxy-arp** is used by routers to answer to ARP queries that are outside the network (enabled by default)
- The **ip local-proxy-arp** is used by routers to answer to ARP queries that are inside the local subnet (disabled by default)
  - The ip local-proxy arp needs ip proxy-arp active in order to function

# Voice VLAN & IP Phones

### Cisco IP Phone

Cisco phones contain an integrated 3-port switch:

- One port connects to the network switch
- One port connects to the end-station (the phone access port)
- One internal port that carries the phone traffic (phone ASIC)




CDP & Cisco Phones

You can leverage CDP to instruct a (cisco) phone to send its traffic using a certain VLAN and CoS value:

- Using the voice VLAN tagged with a L2 CoS value
- Using the access VLAN tagged with a L2 CoS value
- Using the access VLAN untagged with a L2 CoS value

The default CoS value for voice traffic is 5
The default CoS value for call signaling is 3

CDP Extend Priority

You can also leverage CDP to instruct the (cisco) phone to alter the CoS value of traffic arriving on the phone access port (pc)

- Trusted mode - The phone does not change the CoS value of traffic arriving from the PC
- Untrusted mode - The phone rewrites the traffic arriving from the PC to CoS 0 (this is the default)

```
interface
 cdp enable
 switchport priority extend cos cos_value (0-7) | trust
```

**priority extend cos** - Instructs the phone to rewrite the incoming PC traffic (on the phone access port) to this CoS value

- The default is to rewrite all traffic arriving from the PC to CoS 0 (default priority)

**priority extend trust** - Instructs the phone to not alter the CoS value of the incoming PC traffic and blindly forward it to the switch

Do no rewrite traffic arriving from the PC on the phone

```
interface gi1/0/1
 switchport mode dynamic auto
 switchport access vlan 10
 switchport voice vlan 60

 cdp enable
 switchport priority extend trust
```

```
show interfaces gi1/0/1 switchport

Name: Gi1/0/1
Administrative Mode: dynamic auto
Operational Mode: static access
Access Mode VLAN: 10 (VLAN0010)
Voice VLAN: 60 (VLAN0060)

Appliance trust: trusted
```

## Voice VLAN

- You can only configure the voice VLAN on (operational) access ports
  - You don't have to statically force it to be an access port, DTP works as well
  - Technically you can configure it on a trunk, but it will not be operational
  - You cannot configure it on a private-vlan (host or promiscuous) port
  - The voice VLAN is disabled by default
- The voice VLAN is an actual (regular) vlan and needs to be present and active on the switch
- Spanning Tree PortFast will be enabled automatically when you define the voice VLAN
  - PortFast will not be disabled if the voice VLAN configuration is removed
- Generally, the phone will send tagged traffic using the voice VLAN, the PC will send untagged traffic using access VLAN
  - The default CoS value for voice traffic arriving tagged at the voice VLAN is 5
  - The default CoS value for data traffic arriving untagged at the data (regular VLAN) is 0

```
interface
 switchport mode access | dynamic auto
```

switchport voice **vlan** *vlan_id* | *dot1p* | *untagged* | *none*
switchport voice **detect** cisco-phone **full-duplex**

**detect cisco phone** - Detects and recognizes a cisco phone

- If the switch detects another device plugged into this port it will generate a syslog message
- The switch will NOT err-disable the port if another device is plugged in

**detect cisco phone full-duplex** - Detects and recognizes a cisco phone running on a full-duplex link

- If the switch detects another device plugged into this port it will generate a syslog message
- If the switch detects a cisco-phone running half-duplex it will generate a syslog message
- The switch will NOT err-disable the port if another device is plugged in, or if the cisco phone is running half-duplex

Voice VLAN Options

**vlan vlan_id** - Defines the voice VLAN-id (range 1-4094)

- This VLAN is an actual (regular) vlan and needs to be present and active on the switch
- Voice traffic arriving on this port from the phone will be automatically placed in the voice VLAN and forwarded with CoS 5

**vlan dot1p** - Does not define a separate VLAN for voice traffic, instead the access VLAN is shared between the phone and the PC

- Use a single VLAN for data and voice traffic but the 802.1p CoS tag is added to the voice traffic arriving from the phone
  - Voice traffic will receive the 802.1p CoS value of 5
  - Voice signaling (call control) will receive the 802.1p CoS value of 3
  - PC traffic will receive the default CoS value of 0
- Workings of **vlan dot1p** and **vlan** *vlan_id* are very similar, the main difference is that no separate VLAN is needed for dot1p

**vlan none** - Allow the phone to use its own configuration to mark voice traffic and data traffic originated on the PC

- No CoS values are forced on the switch and nothing is remarked
  - Voice traffic will receive the 802.1p CoS value of 5
  - Voice signaling (call control) will receive the 802.1p CoS value of 3
  - PC traffic will receive the default CoS value of 0
- The outcome is very similar to **vlan dot1p**, where no separate VLAN is needed for the voice traffic
- The main difference is that all decisions are left up to the phone

**vlan untagged** - Forces the phone to seed untagged voice traffic using the native VLAN

| **Voice VLAN mode** | **Separate voice VLAN** | **Voice tagged** | **Data tagged** |
|---|---|---|---|
| VLAN-id | yes | yes | no |

| | | | |
|---|---|---|---|
| dot1p | no | yes (dot1p) | yes (dot1p) |
| none | no | yes (dot1p) | yes (dot1p) |
| untagged | no | untagged native VLAN | untagged native VLAN |

Voice VLAN-ID

```
interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 switchport voice vlan 60
```

```
show interfaces gi1/0/1 switchport

Name: Gi1/0/1
Administrative Mode: static access
Operational Mode: static access
Access Mode VLAN: 10 (VLAN0010)
Voice VLAN: 60 (VLAN0060)
```

Voice VLAN dot1p (802.1p)

```
interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 switchport voice vlan dot1p
```

```
show interfaces gi1/0/1 switchport

Name: Gi1/0/1
Administrative Mode: static access
Operational Mode: static access
Access Mode VLAN: 10 (VLAN0010)
Voice VLAN: dot1p
```

Voice VLAN None

```
interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 switchport voice vlan none
```

```
show interfaces gi1/0/1 switchport

Name: Gi1/0/1
Administrative Mode: static access
Operational Mode: static access
Access Mode VLAN: 10 (VLAN0010)
Voice VLAN: none
```

Voice VLAN Untagged

```
interface gi1/0/1
 switchport mode access
 switchport access vlan 10
 switchport voice vlan untagged
```

## LLDP Voice Vlan / Network Policy

- Uses network policy profiles to set the voice VLAN-id and the CoS values
- Also uses LLDP-MED to signal the phone of the used values
- Default DSCP value is EF (46) when configuring the Voice VLAN under the network-policy profile

```
network-policy profile 1
 voice vlan 60 cos 5
 voice vlan 60 dscp 40
 voice vlan dot1p cos 5
 voice vlan none
 voice vlan untagged

int fa0/0
 switchport mode access
 switchport access vlan 10
 network-policy profile 1
 lldp med-tlv-select network-policy

show network-policy profile 1
```

# VTP

## VLAN Trunking Protocol (VTP)

- When the domain is NULL (default) no VTP messages will be sent
    - The switch is waiting for the first reception of a VTP message with a domain name set

- VTP only operates on trunk ports and the VTP domain name is also carried in DTP messages

## VTP (VLAN) Database

- The VLAN database (vlan.dat) is stored in NVRAM
- If during startup the VTP mode is detected as transparent, the VLANs are loaded from the startup-config and the VLAN database contents are ignored
- Does not merge VLAN databases and only overwrites by comparing the MD5 hash
    - The highest revision # will win and overrides the other switches database (see Pruning / Auth / Misc)
    - Devices with same revision number will not update each other when first coming online, because the hash is different
    - Fix this by creating and deleting a VLAN which will increment the revision # by 2
    - VTP authentication alters the MD5 hash of the database (see Pruning / Auth / Misc)
    - The VLAN database name can be changed with the **vtp file** command, this does not create a new database and will only change the name

**vtp file** *filename*

vtp file newvlan.dat

```
dir
Directory of unix:/
786489  -rw-         796  Dec 29 2016 09:25:58 +00:00  newvlan.dat
```

## VTP Modes

| VTP Mode | Create VLANs | Update Database | VLANs Stored In | Purpose |
|---|---|---|---|---|
| Server | yes | yes | VLAN database | Create/delete VLANs and advertise VTP information |
| Client | no | yes | VLAN database | Listen for and forward VTP advertisements<br>Relay VTP information |
| Transparent | yes | no | Running-config | Does not participate in VTP and only relays information<br>VTPv1 relay requires same domain name and revision #<br>VTPv2/3 will relay regardless of local VTP configuration |

| | | | | |
|---|---|---|---|---|
| Off | yes | no | Running-config | Does not participate in VTP and does not relay information |

VTP off mode functions the same as transparent mode except that VTP advertisements are not forwarded

## VTP Versions

| Version | Default | Extended VLAN Range | Authentication | Private VLANs / RSPAN | Primary / Secondary servers |
|---|---|---|---|---|---|
| VTPv1 | yes | no | basic / md5 | no | no |
| VTPv2 | no | no | basic / md5 | no | no |
| VTPv3 | no | yes | enhanced / md5 | yes | yes |

VTPv1 and v2 are designed to work with ISL, this is why extended range VLANs are not supported

- The VLAN extended range is 1006-4096
- Extended VLANs can only be configured with VTPv3 or VTPv2 transparent mode
    - Stored in the running config using VTPv1/v2, not the VLAN database file
    - Lost if VTPv1/2 mode is changed back to server from transparent

## VTP Advertisements

| Type | Contains | Sent by | Reason / Purpose | Description |
|---|---|---|---|---|
| Summary | Version<br>Domain<br>Revision #<br>Time stamp<br>MD5 hash<br># of subset | Server<br>Client | Change in VLAN database<br>Every 300 seconds<br>Response to Request | Advertises general VTP information and how many subset advertisements will follow |
| Subset | VLAN creation, removal. change | Server<br>Client | Change in VLAN database | Advertises the complete VLAN list present on the switch<br>Also advertised when VLAN changes are applied such as type, MTU, name, etc. |
| Request | Version<br>Domain | Server<br>Client | Update request<br>Synchronization | Switch requests update from server after joining the VTP domain<br>Can also be sent if a switch detects (from a summary) that its revision # is lower, in this case the server will respond with a |

| | | | | |
|---|---|---|---|---|
| | | | | new summary + subset advertisement<br>This generally happens when VTP domain or mode is changed (resets revision # to 0) |
| Join | Pruning VLAN list | All | Pruning | The switch advertises which VLANs are needed on the switch |

```
vtp mode server
vtp version 1
vtp domain cisco
```

- The VTP database checksum can mismatch if both servers are at revision 0
    - Create a VLAN to update both switches to revision 1

```
show vtp interface

Interface                VTP Status
-----------------------------------
GigabitEthernet1/0/1      enabled
GigabitEthernet1/0/2      enabled
GigabitEthernet1/0/3      enabled
GigabitEthernet1/0/4      enabled
```

```
show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : aabb.cc00.0100
Configuration last modified by 0.0.0.0 at 12-29-16 09:22:07
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
--------------
VTP Operating Mode                : Server
Maximum VLANs supported locally   : 1005
Number of existing VLANs          : 8
Configuration Revision            : 3
MD5 digest                        : 0x9B 0x76 0xCD 0x67 0x7B 0xF2 0xD8 0x11
                                    0x77 0xED 0x80 0xE5 0xCA 0x0F 0xAA 0xC7
```

# VTPv3

## VTPv3

- Backwards compatible with v2 and will send VTPv3 and v2 packets to devices over a trunk port
- Devices supporting VTPv1/2 will automatically update to v2 when communicating with v3 neighbors
    - Exception is when extended range VLANs are used in the VTPv3 domain
    - Changes should be made on v3 switches with the v2 devices configured as clients
    - If changes are made on the v2 devices, only v2 neighbors will update their revision numbers
    - Updates from v3 neighbors will be rejected later on, because of the lower revision number

## VTPv3 Enhancements

- Supports extended VLAN range (1006-4096), private VLANs and Remote SPAN VLANs
- Enhanced authentication, ability to hide password in running-config
- Supports separate databases (MST and VLAN)

## VTPv3 Server Modes

| **Server Mode** | **Purpose** |
|---|---|
| Primary | Can modify VTP, only 1 per domain |
| Secondary | Cannot modify VTP, can be promoted to primary |

Primary is the only switch in a VTPv3 domain whose VLAN database can be propagated

- Can be demoted by changing modes or by configuring a VTP password
- Different servers can be primary for the MST / VLAN databases
    - The MST database is the instance to VLAN mapping, not the actual VLAN list
- If the VTP password is configured, you need to re-enter it when promoting a server

Server

```
vtp version 3
vtp mode server
vtp domain cisco
vtp password cisco hidden
vtp primary vlan force
debug sw-vlans vtp events
```

**force** - the switch does not check for conflicting devices (other primary servers)
**hidden** - hides the password in the running-config

Client

```
vtp version 3
vtp mode client
vtp domain cisco
vtp password cisco hidden
```

```
show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 3
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : aabb.cc00.0100

Feature VLAN:
--------------
VTP Operating Mode                : Primary Server
Number of existing VLANs          : 10
Number of existing extended VLANs : 0
Maximum VLANs supported locally   : 4096
Configuration Revision            : 3
Primary ID                        : aabb.cc00.0100
Primary Description               : CSW1
MD5 digest                        : 0x7C 0x49 0x3D 0xB3 0x77 0xD3 0xFA 0x40
                                    0x36 0x5E 0xBA 0xCE 0xEB 0x6E 0xF9 0xE1


Feature MST:
--------------
VTP Operating Mode                : Transparent


Feature UNKNOWN:
--------------
VTP Operating Mode                : Transparent
```

```
show vtp devices
Retrieving information from the VTP domain. Waiting for 5 seconds.

VTP Feature  Conf Revision Primary Server Device ID      Device Description
------------ ---- -------- -------------- -------------- --------------------
VLAN         No   3        aabb.cc00.0100=aabb.cc00.0100 CSW1
```

```
show vtp password
VTP Password: 52C518C46CE6647155DE01304CECA63E
```

# Pruning / Auth / Misc

## VTP Synchronization / Revision Number Issues (Bomb)

- The VTP revision number is stored in NVRAM and is not altered by a power cycle
- The revision # can be reset to 0 only by using one of the following methods:
  - Change the mode to transparent and then change the mode back to server (VTPv1/2 only)
  - Change the VTP domain
  - Configure a VTP password
- Issues occur when a VLAN databases is overwritten by clients/servers with a higher revision #
  - This happens regardless of the amount of VLANs present, only the revision # matters
  - Even VTP clients can override the database of servers in VTPv1/2/3

- A VTP client can update the VLAN database if the following is true:
  - The new link connecting the new switch is trunking
  - The new switch has the same VTP domain name / password as the other switches.
  - The new switch's revision number is higher than that of the existing switches
- Even in VTPv3 this can happen if the switches agree on the identity of the primary server

Revision numbers are incremented based on individual VLAN configuration statements

Will increment the revision number by 1

```
vlan 100-1000
```

Will increment the revision number by 3

```
vlan 100
vlan 200
vlan 300
```

## VTP Authentication

- VTP authentication re-hashes the entire VLAN database MD5 hash, not just the password string
- VTPv1/2 password is stored in clear text in the running config
- VTPv3 password can be hidden from the running config with the **hidden** keyword
    - After configuring a VTPv3 password it is required in order to promote a server to primary
    - The hidden password feature is not backwards compatible with v2 neighbors

**vtp password** *value* **hidden**

```
vtp password cisco
```

```
show vtp password
VTP Password: cisco
```

## VTP Pruning

- VTP pruning is an extension of VTPv1 and is supported on all versions (disabled by default)
- Via VTP join messages switches communicate which VLANs it only need frames for
    - This is decided based on active trunk/access ports in a specific VLAN
    - The VLANs are still added to the database, however they won't be active on the trunk
- Unlike manual pruning, VTP pruning does NOT limit the amount of active STP instances (in case of RPVST+/PVST+)
    - VTP pruning works alongside STP and allows pruned VLANs in the management domain, but not in the STP forwarding state
    - Manual pruning does not allow pruned VLANs in the management domain
- Can only be enabled on servers, only works on servers and clients
- By default all normal-range VLANs will be pruned (except 1 / 1002-1005)
    - VTP pruning only works for VLANs 2-1001 (even with VTPv3)
    - Control which VLANs are pruned with the **switchport trunk pruning vlan** command
    - VLANs specified here will be eligible for pruning, anything not specified will not be pruned
    - Verify with **show interfaces trunk**

**switchport trunk pruning vlan** *[ add | except | remove | vlan-list ] none*

Server

```
vtp mode server
vtp pruning
vlan 10,20,30,40,50
```

Client

```
interface gi1/0/1
 switchport trunk pruning vlan 10-30,50
```

```
show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi1/0/1     desirable        n-isl          trunking      1
Port        Vlans allowed on trunk
Gi1/0/1     1-4094
Port        Vlans allowed and active in management domain
Gi1/0/1     1,10,20,30,40,50
Port        Vlans in spanning tree forwarding state and not pruned
Gi1/0/1     1,40
```

## Manual Pruning vs VTP pruning

- VTP pruning only works for VLANs 2-1001 (even with VTPv3)
- You can manually prune VLAN1
- VTP pruning allows the VLAN on the link and allows the VLAN to participate in STP
- Manual pruning denies the VLAN on the link and does not allow it to participate in STP

VTP pruning

```
vtp mode server
vtp domain cisco
vtp pruning

vlan 10,20,30,40,50

interface gi1/0/0
 switchport mode trunk
 switchport trunk encapsulation dot1q

interface gi1/0/1
 switchport access vlan 10

interface gi1/0/2
 switchport access vlan 20

interface gi1/0/3
 switchport access vlan 30
```

```
show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi1/0/0     on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi1/0/0     1-4094

Port        Vlans allowed and active in management domain
Gi1/0/0     1,10,20,30,40,50

Port        Vlans in spanning tree forwarding state and not pruned
Gi1/0/0     1,10,20,30
```

- In the output you can see that VLANs 1,10,20,30,40,50 are allowed and active in the management domain
    - However only 1,10,20,30 are in the STP forwarding state
    - This area is where VTP pruning takes effect
    - If manual pruning was configured, only 1,10,20,30 would be allowed and active in the management domain

```
show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi1/0/0     on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi1/0/0     1,10,20,30

Port        Vlans allowed and active in management domain
Gi1/0/0     1,10,20,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi1/0/0     1,10,20,30
```

Manual pruning

```
vtp mode off

vlan 10,20,30,40,50

interface gi1/0/0
 switchport mode trunk
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,10,20,30

interface gi1/0/1
 switchport access vlan 10
```

```
interface gi1/0/2
 switchport access vlan 20

interface gi1/0/3
 switchport access vlan 30
```

## Disable VTP

- Not supported on all platforms
- If not supported, 'turn off' VTP with mode transparent
- Disable globally or on an interface basis

```
vtp mode off
int gi1/0/1
 no vtp
```

## VTPv3 Feature Unknown

- Allows you to configure the behavior of the switch databases that it cannot (yet) interpret
    - These databases will be features handled by future extensions of VTP version 3
    - Default is off for unknown instances, can set to transparent

```
vtp mode off unknown
vtp mode transparent unknown
```