UTP vs Fiber-Optic Cabling

UTP

- Lower cost than fiber-optic.
- Shorter maximum distance than fiber-optic (~100m).
- Can be vulnerable to EMI (Eletromagnetic Interference).
- RJ45 ports used with UTP are cheaper than SFP ports.
- Emit (leak) a faint signal outside of the cable, which can be copied (=security risk)

Fiber-Optic

- Higher cost than UTP.
- Longer maximum distance than UTP.
- No vulnerability to EMI.
- SFP ports are more expensive than RJ45 ports (single-mode is more expensive than multimode).
- Does not emit any signal outside of the cable (=no security risk).

show ip interface brief

```
SW1>en
SW1#sh ip int br
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/9        unassigned      YES unset  down                  down
FastEthernet0/10       unassigned      YES unset  down                  down
FastEthernet0/11       unassigned      YES unset  down                  down
FastEthernet0/12       unassigned      YES unset  down                  down
```

Router interfaces have the `shutdown` command applied by default
=will be in the `administratively down/down` state by default

Switch interfaces do NOT have the '`shutdown`' command applied by default
=will be in the `up/up` state if connected to another device
OR
in the `down/down` state if not connected to another device

CSMA/CD

- **C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **D**etection
- Before sending frames, devices 'listen' to the collision domain until they detect that other devices are not sending.
- If a collision does occur, the device sends a jamming signal to inform the other devices that a collision happened.
- Each device will wait a random period of time before sending frames again.
- The process repeats.

Full/Half Duplex

- **Half duplex**: The device cannot send and receive data at the same time. If it is receiving a frame, it must wait before sending a frame.
- Devices attached to a **hub** must operate in half duplex.

- **Full duplex**: The device **can** send and receive data at the same time. It does not have to wait.

Speed/Duplex Autonegotiation

- What if autonegotiation is disabled on the device connected to the switch?
- **SPEED**: The switch will try to sense the speed that the other device is operating at.
  If if fails to sense the speed, it will use the slowest supported speed (ie. 10 Mbps on a 10/100/1000 interface)
- **DUPLEX**: If the speed is 10 or 100 Mbps, the switch will use half duplex.
  If the speed is 1000 Mbps or greater, use full duplex.

Interface Errors

```
269 packets input, 71059 bytes, 0 no buffer
Received 6 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
7290 packets output, 429075 bytes, 0 underruns
0 output errors, 3 interface resets
0 output buffer failures, 0 output buffers swapped out
```

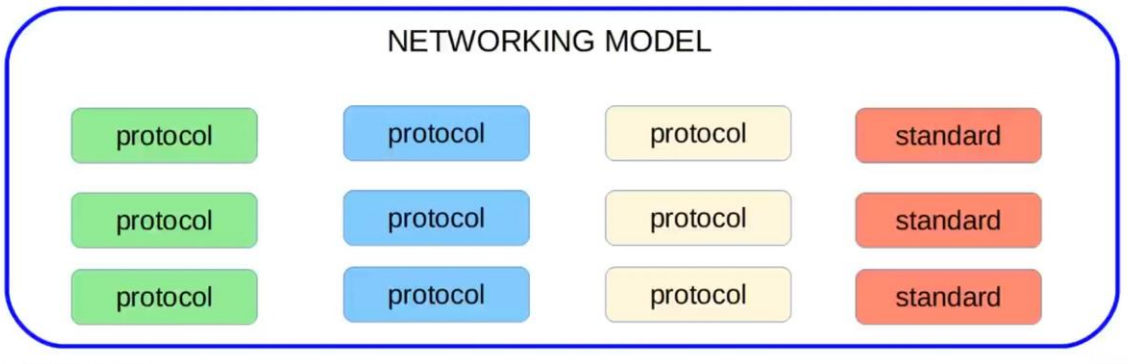
- **Runts**: Frames that are smaller than the minimum frame size (64 bytes)
- **Giants**: Frames that are larger than the maximum frame size (1518 bytes)
- **CRC**: Frames that failed the CRC check (in the Ethernet FCS trailer)
- **Frame**: Frames that have an incorrect format (due to an error)
- **Input errors**: Total of various counters, such as the above four
- **Output errors**: Frames the switch tried to send, but failed due to an error

What is a networking model?

Networking models categorize and provide a structure for networking protocols and standards.

logical

A set of rules defining how network devices and software should work.



OSI Model

7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

- 'Open Systems Interconnection' model
- A conceptual model that categorizes and standardizes the different functions in a network.
- Created by the 'International Organization for Standardization' (ISO).
- Functions are divided into 7 'Layers'.
- These layers work together to make the network work.

TCP/IP Suite

- Conceptual model and set of communications protocols used in the Internet and other networks.
- Known as TCP/IP because those are two of the foundational protocols in the suite.
- Developed by the United States Department of Defense through DARPA (Defense Advanced Research Projects Agency)
- Similar structure to the OSI Model, but with fewer layers.
- This is the model actually in use in modern networks.
- NOTE: The OSI model still influences how network engineers think and talk about networks.

service password-encryption

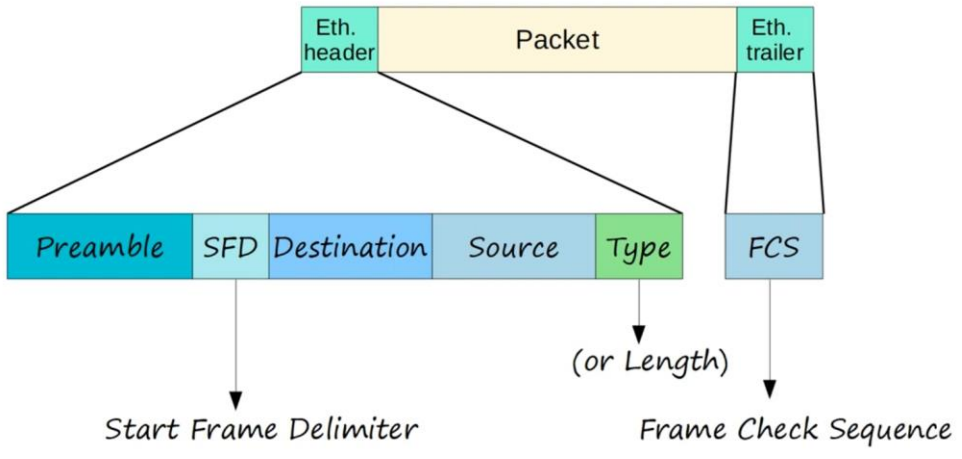
If you enable **service password-encryption**...

- current passwords **will** be encrypted.
- future passwords **will** be encrypted.
- the **enable secret** will not be effected.

If you disable **service password-encryption**...

- current passwords **will not** be decrypted.
- future passwords **will not** be encrypted.

Ethernet Frame



MAC Address

- 6-byte (48-bit) physical address assigned to the device when it is made
- A.K.A. 'Burned-In Address' (BIA)
- Is globally unique
- The first 3 bytes are the OUI (Organizationally Unique Identifier), which is assigned to the company making the device
- The last 3 bytes are unique to the device itself

ARP

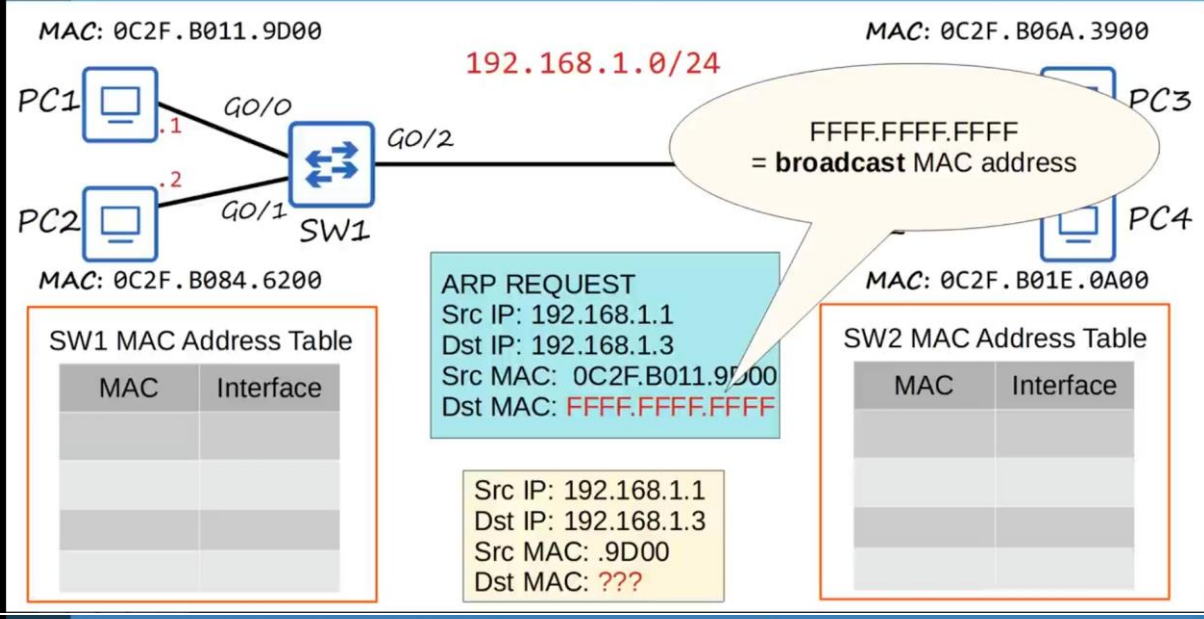
- ARP stands for 'Address Resolution Protocol'
- ARP is used to discover the Layer 2 address (MAC address) of a known Layer 3 address (IP address)
- Consists of two messages:

  ARP Request

  ARP Reply

- **ARP Request** is **broadcast** = sent to all hosts on the network
- **ARP Reply** is **unicast** = sent only to one host (the host that sent the request)

ARP Request



Topics we covered

- Ethernet frame payload minimum size
- ARP (Address Resolution Protocol)

  ARP Request

  ARP Reply

- ARP table
- Ping

  ICMP Echo Request

  ICMP Echo Reply

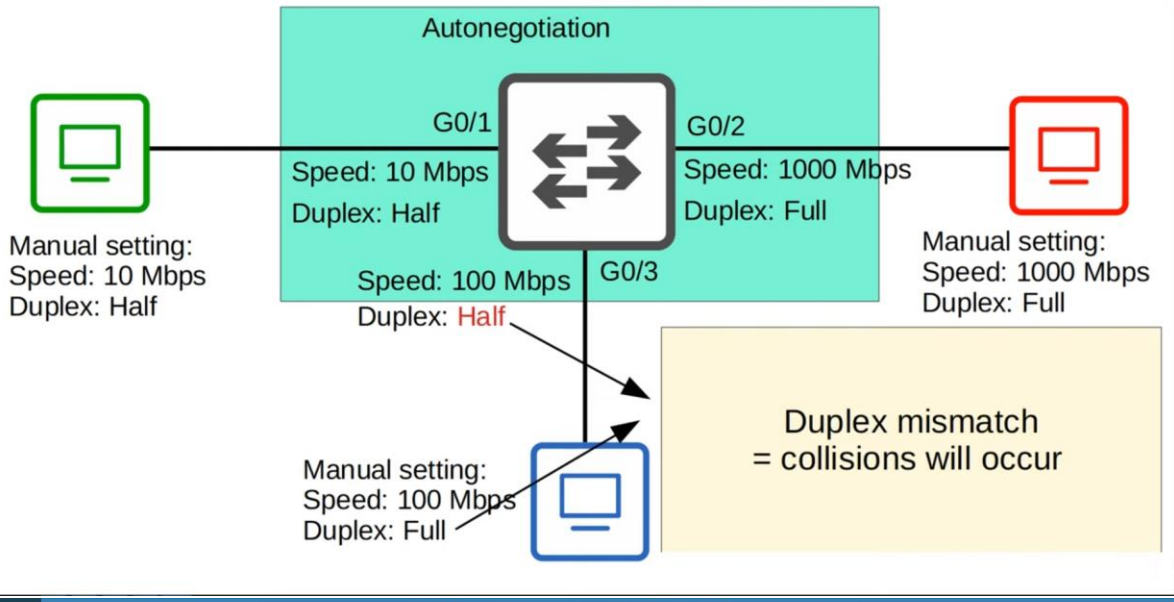
- MAC Address Table

Quiz Question 4

★a) Broadcast, unknown unicast

Broadcast frames have a destination address of FFFF.FFFF.FFFF and are sent to all hosts on the local network.
Unknown unicast frames are destined for a single host, however the switch doesn't have an entry for the destination in its MAC address table so it must flood the frame.

Speed/Duplex Autonegotiation



What are ACLs?

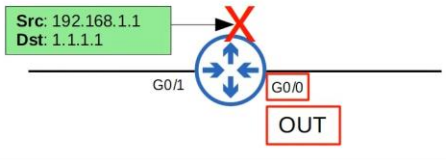
- ACLs (Access Control Lists) have multiple uses.
- In Day 34 and Day 35, we will focus on ACLs from a security perspective.
- ACLs function as a packet filter, instructing the router to permit or discard specific traffic.
- ACLs can filter traffic based on source/destination IP addresses, source/destination Layer 4 ports, etc.

How ACLs work

- Configuring an ACL in global config mode will not make the ACL take effect.
- The ACL must be applied to an interface.
- ACLs are applied either inbound or outbound.
- ACLs are made up of one or more ACEs.

A maximum of one ACL can be applied to a single interface per direction.
Inbound: Maximum one ACL
Outbound: Maximum one ACL

- When the router checks a packet against the ACL, it processes the ACEs in order, from top to bottom.
- If the packet matches one of the ACEs in the ACL, the router takes the action and stops processing the ACL. All entries below the matching entry will be ignored.



ACL 2:
1: if source IP = 192.168.0.0/16, then deny
~~2: if source IP = 192.168.1.0/24, then permit~~

ACL Types

- Standard ACLs: Match based on **Source IP address** only
  - → Standard Numbered ACLs
  - → Standard Named ACLs

- Extended ACLs: Match based on **Source/Destination IP, Source/Destination port,** etc.
  - → Extended Numbered ACLs
  - → Extended Named ACLs

Standard Numbered ACLs

- Standard ACLs match traffic based only on the source IP address of the packet.
- Numbered ACLs are identified with a number (ie. ACL 1, ACL 2, etc)
- Different types of ACLs have a different range of numbers that can be used.
  → Standard ACLs can use 1-99 and 1300-1999.
- The basic command to configure a standard numbered ACL is:
  `R1(config)# access-list number {deny | permit} ip wildcard-mask`

```
R1(config)# access-list 1 deny 1.1.1.1 0.0.0.0
R1(config)# access-list 1 deny 1.1.1.1
R1(config)# access-list 1 deny host 1.1.1.1
R1(config)# access-list 1 permit any
R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255
R1(config)# access-list 1 remark ## BLOCK BOB FROM ACCOUNTING ##
```

Standard Named ACLs

- Standard ACLs match traffic based only on the source IP address of the packet.
- Named ACLs are identified with a name (ie. 'BLOCK_BOB')
- Standard named ACLs are configured by entering 'standard named ACL config mode', and then configuring each entry within that config mode.
  `R1(config)# ip access-list standard acl-name`
  `R1(config-std-nacl)# [entry-number] {deny | permit} ip wildcard-mask`

```
R1(config)#ip access-list standard BLOCK_BOB
R1(config-std-nacl)#5 deny 1.1.1.1
R1(config-std-nacl)#10 permit any
R1(config-std-nacl)#remark ## CONFIGURED NOV 21 2020 ##
R1(config-std-nacl)#interface g0/0
R1(config-if)#ip access-group BLOCK_BOB in
```

Configuring numbered ACLs with subcommands

- In Day 34, you learned that numbered ACLs are configured in global config mode:

```
R1(config)# access-list 1 deny 192.168.1.1
R1(config)# access-list 1 permit any
```

- You learned that named ACLs are configured with subcommands in a separate config mode:

```
R1(config)# ip access-list standard BLOCK_PC1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

- However, in modern IOS you can also configure numbered ACLs in the exact same way as named ACLs:

```
R1(config)# ip access-list standard 1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

- This is just a different way of configuring numbered ACLs. However, in the running-config the ACL will display as if it was configured using the traditional method.

Extended ACLs

- Extended ACLs function mostly the same as standard ACLs.
- They can be numbered or named, just like standard ACLs.
  →Numbered ACLs use the following ranges: 100 – 199, 2000 – 2699
- They are processed from top to bottom, just like standard ACLs.
- However, the can match traffic based on more parameters, so they are more precise (and more complex) than standard ACLs.
- We will focus on matching based on these main parameters: **Layer 4 protocol/port**, **source address**, and **destination address**.

`R1(config)# access-list number [permit | deny] protocol src-ip dest-ip`

`R1(config)# ip access-list extended {name | number}`
`R1(config-ext-nacl)# [seq-num] [permit | deny] protocol src-ip dest-ip`

Extended ACL entry practice (1)

1. Allow all traffic

```
R1(config-ext-nacl)#permit ip any any
```

2. Prevent 10.0.0.0/16 from sending UDP traffic to 192.168.1.1/32

```
R1(config-ext-nacl)#deny udp 10.0.0.0 0.0.255.255 host 192.168.1.1
```

3. Prevent 172.16.1.1/32 from pinging hosts in 192.168.0.0/24

```
R1(config-ext-nacl)#deny icmp host 172.16.1.1 192.168.0.0 0.0.0.255
```

Matching the TCP/UDP port numbers

- When matching TCP/UDP, you can optionally specify the source and/or destination port numbers to match.

```
R1(config-ext-nacl)#deny tcp src-ip  eq    src-port-num   dest-ip  eq    dst-port-num
                                     gt                             gt
                                     lt                             lt
                                     neq                            neq
                                     range                          range
```

- **eq 80** = equal to port 80
- **gt 80** = greater than 80 (81 and greater)
- **lt 80** = less than 80 (79 and less)
- **neq 80** = NOT 80
- **range 80 100** = from port 80 to port 100

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

Matching the TCP/UDP port numbers

```
R1(config-ext-nacl)#deny tcp any host 1.1.1.1 eq ?
  <0-65535>     Port number
  bgp           Border Gateway Protocol (179)
  chargen       Character generator (19)
  cmd           Remote commands (rcmd, 514)
  daytime       Daytime (13)
  discard       Discard (9)
  domain        Domain Name Service (53)
  drip          Dynamic Routing Information Protocol (3949)
  echo          Echo (7)
  exec          Exec (rsh, 512)
  finger        Finger (79)
  ftp           File Transfer Protocol (21)
  ftp-data      FTP data connections (20)
  gopher        Gopher (70)
  hostname      NIC hostname server (101)
  ident         Ident Protocol (113)
  irc           Internet Relay Chat (194)
  klogin        Kerberos login (543)
  kshell        Kerberos shell (544)
  login         Login (rlogin, 513)
  lpd           Printer service (515)
  nntp          Network News Transport Protocol (119)
  onep-plain    ONEP Cleartext (15001)
  onep-tls      ONEP TLS (15002)
  pim-auto-rp   PIM Auto-RP (496)
  pop2          Post Office Protocol v2 (109)
  pop3          Post Office Protocol v3 (110)
  smtp          Simple Mail Transport Protocol (25)
  sunrpc        Sun Remote Procedure Call (111)
  tacacs        TAC Access Control System (49)
  talk          Talk (517)
  telnet        Telnet (23)
  time          Time (37)
  uucp          Unix-to-Unix Copy Program (540)
  whois         Nicname (43)
  www           World Wide Web (HTTP, 80)
```

```
R1(config-std-nacl)#deny tcp any host 1.1.1.1 eq 80
```

→ Deny all packets destined for IP address 1.1.1.1/32, TCP port 80.

After the destination IP address and/or destination port numbers, there are many more options you can use to match (not necessary for the CCNA).
Some examples:

- **ack**: match the TCP ACK flag
- **fin**: match the TCP FIN flag
- **syn**: match the TCP SYN flag
- **ttl**: match packets with a specific TTL value
- **dscp**: match packets with a specific DSCP value

Extended ACL entry practice (2)

1. Allow traffic from 10.0.0.0/16 to access the server at 2.2.2.2/32 using HTTPS.

```
R1(config-ext-nacl)#permit tcp 10.0.0.0 0.0.255.255 2.2.2.2 0.0.0.0 eq 443
```

2. Prevent all hosts using source UDP port numbers from 20000 to 30000 from accessing the server at 3.3.3.3/32.

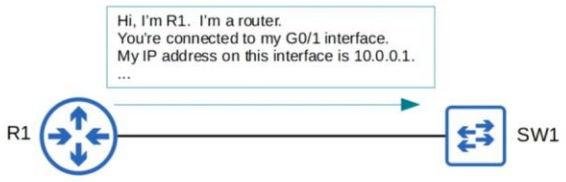
```
R1(config-ext-nacl)#deny udp any range 20000 30000 host 3.3.3.3
```

3. Allow hosts in 172.16.1.0/24 using a TCP source port greater than 9999 to access all TCP ports on server 4.4.4.4/32 except port 23.

```
R1(config-ext-nacl)#permit tcp 172.16.1.0 0.0.0.255 gt 9999 host 4.4.4.4 neq 23
```

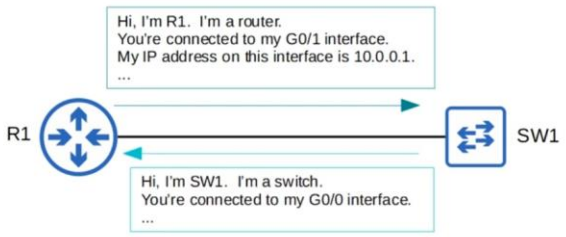
Layer 2 Discovery Protocols

- Layer 2 discovery protocols such as CDP and LLDP share information with and discover information about neighboring (connected) devices.
- The shared information includes host name, IP address, device type, etc.
- CDP is a Cisco proprietary protocol.
- LLDP is an industry standard protocol (IEEE 802.1AB).
- Because they share information about the devices in the network, they can be considered a security risk and are often not used. It is up to the network engineer/admin to decide if they want to use them in the network or not.



Layer 2 Discovery Protocols

- Layer 2 discovery protocols such as CDP and LLDP share information with and discover information about neighboring (connected) devices.
- The shared information includes host name, IP address, device type, etc.
- CDP is a Cisco proprietary protocol.
- LLDP is an industry standard protocol (IEEE 802.1AB).
- Because they share information about the devices in the network, they can be considered a security risk and are often not used. It is up to the network engineer/admin to decide if they want to use them in the network or not.



Cisco Discovery Protocol

- CDP is a Cisco proprietary protocol.
- It is enabled on Cisco devices (routers, switches, firewalls, IP phones, etc) by default.
- CDP messages are periodically sent to multicast MAC address `0100.0CCC.CCCC`.
- When a device receives a CDP message, it processes and discards the message. It does NOT forward it to other devices.
- By default, CDP messages are sent once every **60 seconds**.
- By default, the CDP holdtime is **180 seconds**. If a message isn't received from a neighbor for 180 seconds, the neighbor is removed from the CDP neighbor table.
- CDPv2 messages are sent by default.

CDP show commands summary

- `R1#` **`show cdp`**
  → shows basic information about CDP (timers, version)
- `R1#` **`show cdp traffic`**
  → displays how many CDP messages have been sent and received
- `R1#` **`show cdp interface`**
  → displays which interfaces CDP is enabled on
- `R1#` **`show cdp neighbors`**
  → lists CDP neighbors and some basic information about each neighbor
- `R1#` **`show cdp neighbors detail`**
  → lists each CDP neighbor with more detailed information
- `R1#` **`show cdp entry`** *name*
  → displays the same info as above, but for the specified neighbor only

CDP Configuration Commands

- CDP is globally enabled by default.
- CDP is also enabled on each interface by default.
- To enable/disable CDP globally: `R1(config)# [no] cdp run`
- To enable/disable CDP on specific interfaces: `R1(config-if)# [no] cdp enable`
- Configure the CDP timer: `R1(config)# cdp timer seconds`
- Configure the CDP holdtime: `R1(config)# cdp holdtime seconds`
- Enable/disable CDPv2: `R1(config)# [no] cdp advertise-v2`

Link Layer Discovery Protocol

- LLDP is an industry standard protocol (IEEE 802.1AB).
- It is usually disabled on Cisco devices by default, so it must be manually enabled.
- A device can run CDP and LLDP at the same time.
- LLDP messages are periodically sent to multicast MAC address `0180.C200.000E`.
- When a device receives an LLDP message, it processes and discards the message. It does NOT forward it to other devices.
- By default, LLDP messages are sent once every **30 seconds**.
- By default, the LLDP holdtime is **120 seconds**.
- LLDP has an additional timer called the 'reinitialization delay'. If LLDP is enabled (globally or on an interface), this timer will delay the actual initialization of LLDP. **2 seconds** by default.

LLDP Configuration Commands

- LLDP is usually globally disabled by default.
- LLDP is also disabled on each interface by default.
- To enable LLDP globally: `R1(config)# lldp run`
- To enable LLDP on specific interfaces (tx): `R1(config-if)# lldp transmit`
- To enable LLDP on specific a interface (rx): `R1(config-if)# lldp receive`
- Configure the LLDP timer: `R1(config)# lldp timer seconds`
- Configure the LLDP holdtime: `R1(config)# lldp holdtime seconds`
- Configure the LLDP reinit timer: `R1(config)# lldp reinit seconds`

The Purpose of DHCP

- DHCP is used to allow hosts to automatically/dynamically learn various aspects of their network configuration, such as IP address, subnet mask, default gateway, DNS server, etc, without manual/static configuration.
- It is an essential part of modern networks.
  → When you connect a phone/laptop to WiFi, do you ask the network admin which IP address, subnet mask, default gateway, etc, the phone/laptop should use?
- Typically used for 'client devices' such as workstations (PCs), phones, etc.
- Devices such as routers, servers, etc, are usually manually configured.
- In small networks (such as home networks) the router typically acts as the DHCP server for hosts in the LAN.
- In larger networks, the DHCP server is usually a Windows/Linux server.

The Basic Functions of DHCP

```
C:\Users\user>ipconfig /all

[output omitted]

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82579LM Gigabit Network Connection
   Physical Address. . . . . . . . . : 78-2B-CB-AC-08-67
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.0.167(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, January 23, 2021 12:02:04 PM
   Lease Expires . . . . . . . . . . : Saturday, January 23, 2021 2:02:05 PM
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]
```

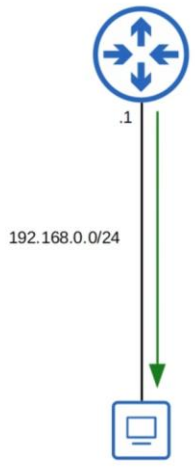
This PC was previously assigned this IP address by the DHCP server, so it asked to receive the same address again this time.

The Basic Functions of DHCP

```
C:\Users\user>ipconfig /all

[output omitted]

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82579LM Gigabit Network Connection
   Physical Address. . . . . . . . . : 78-2B-CB-AC-08-67
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.0.167(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, January 23, 2021 12:02:04 PM
   Lease Expires . . . . . . . . . . : Saturday, January 23, 2021 2:02:05 PM
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]
```

DHCP server 'lease' IP address to clients.
These leases are usually not permanent, and the client must give up the address at the end of the lease.

DHCP Offer




```
No.  Time                 Source          Destination     Protocol  Length  Info
 262 13:27:34.562795      192.168.0.1     192.168.0.167   DHCP      342     DHCP Offer    - Transaction ID 0xd7a1c480

> Frame 262: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface \Device\NPF_{9956EC07-3774-4B11-970C
> Ethernet II, Src: Tp-LinkT_dd:a8:e4 (98:da:c4:dd:a8:e4), Dst: Dell_ac:08:67 (78:2b:cb:ac:08:67)
> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.167
> User Datagram Protocol, Src Port: 67, Dst Port: 68
v Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xd7a1c480
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.0.167
    Next server IP address: 192.168.0.1
    Relay agent IP address: 0.0.0.0
    Client MAC address: Dell_ac:08:67 (78:2b:cb:ac:08:67)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Offer)
  > Option: (54) DHCP Server Identifier (192.168.0.1)
  > Option: (51) IP Address Lease Time
  > Option: (58) Renewal Time Value
  > Option: (59) Rebinding Time Value
  > Option: (1) Subnet Mask (255.255.255.0)
  > Option: (28) Broadcast Address (192.168.0.255)
  > Option: (6) Domain Name Server
  > Option: (3) Router
  > Option: (255) End
    Padding: 000000000000000
```

The DHCP Offer message can be either **broadcast** or **unicast**.

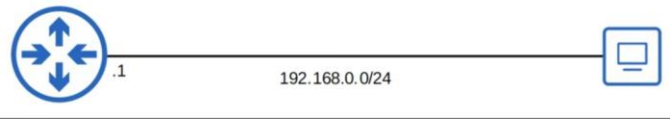
DHCP Ack

DHCP Discover:
Are there any DHCP servers in this network?
I need an IP address.

DHCP Offer:
How about this IP address?

DHCP Request:
I want to use the IP address you offered me.

DHCP Ack:
Okay, you may use it.



DHCP D-O-R-A

Discover	Client → Server	Broadcast
Offer	Server → Client	Broadcast or Unicast
Request	Client → Server	Broadcast
Ack	Server → Client	Broadcast or Unicast

DHCP Server Configuration in IOS

```
R1(config)#ip dhcp excluded-address 192.168.1.1 192.168.1.10
R1(config)#ip dhcp pool LAB_POOL
R1(dhcp-config)#network 192.168.1.0 ?
  /nn or A.B.C.D  Network mask or prefix length
  <cr>
R1(dhcp-config)#network 192.168.1.0 /24
R1(dhcp-config)#dns-server 8.8.8.8
R1(dhcp-config)#domain-name jeremysitlab.com
R1(dhcp-config)#default-router 192.168.1.1
R1(dhcp-config)#lease 0 5 30
```

Specify a range of addresses that **won't** be given to DHCP clients.

Create a DHCP pool.

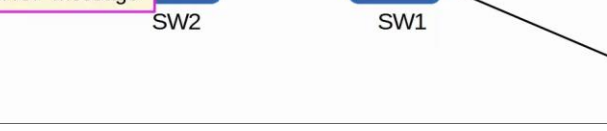
Specify the subnet of addresses to be assigned to clients (except the excluded addresses)

Specify the DNS server that DHCP clients should use.

Specify the domain name of the network. (ie. PC1 = pc1.jeremysitlab.com)

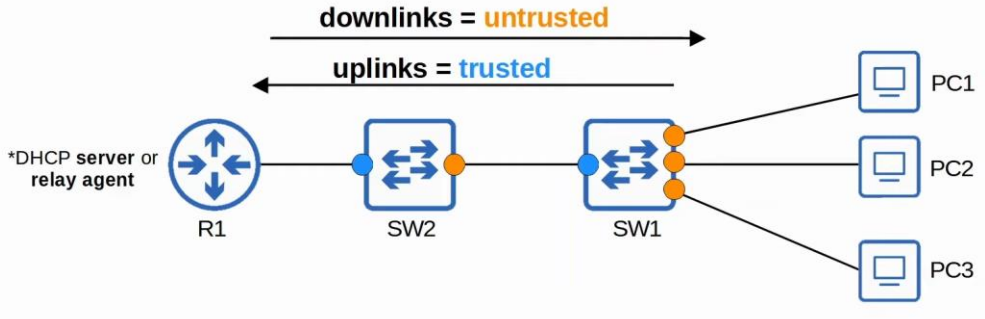
Specify the default gateway.

Specify the lease time.
lease *days hours minutes* OR
lease infinite



DHCP Snooping

- DHCP snooping is a security feature of switches that is used to filter DHCP messages received on *untrusted* ports.
- DHCP snooping only filters DHCP messages. Non-DHCP messages aren't affected.
- All ports are *untrusted* by default.
  →Usually, **uplink** ports are configured as *trusted* ports, and **downlink** ports remain *untrusted.*

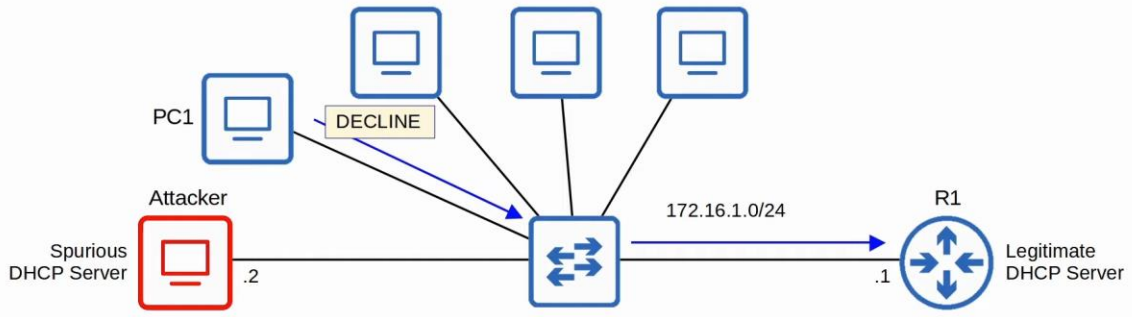


DHCP Snooping

- DHCP snooping is a security feature of switches that is used to filter DHCP messages received on *untrusted* ports.
- DHCP snooping only filters DHCP messages. Non-DHCP messages aren't affected.
- All ports are *untrusted* by default.
  →Usually, **uplink** ports are configured as *trusted* ports, and **downlink** ports remain *untrusted.*

DHCP Poisoning (Man-in-the-Middle)

- Similar to ARP Poisoning, DHCP Poisoning can be used to perform a Man-in-the-Middle attack.
- A spurious DHCP server replies to clients' DHCP Discover messages and assigns them IP addresses, but makes the client use the spurious server's IP as the default gateway.
  *Clients usually accept the first OFFER message they receive.
- This will cause the client to send traffic to the attacker instead of the legitimate default gateway.
- The attacker can then examine/modify the traffic before forwarding it to the legitimate default gateway.



DHCP Messages

- When DHCP Snooping filters messages, it differentiates between **DHCP Server** messages and **DHCP Client** messages

- Messages sent by **DHCP Servers**:
  → OFFER
  → ACK
  → NAK = Opposite of ACK, used to decline a client's REQUEST

- Messages sent by **DHCP Clients:**
  → DISCOVER
  → REQUEST
  → RELEASE = Used to tell the server that the client no longer needs its IP address
  → DECLINE = Used to decline the IP address offered by a DHCP server

DHCP Snooping Operations

- If a DHCP message is received on a **trusted port**, forward it as normal without inspection.

- If a DHCP message is received on an **untrusted port**, inspect it and act as follows:
  → If it is a **DHCP Server** message, discard it.

  → If it is a **DHCP Client** message, perform the following checks:

  DISCOVER/REQUEST messages: Check if the frame's source MAC address and the DHCP message's CHADDR fields match. Match = forward, mismatch = discard

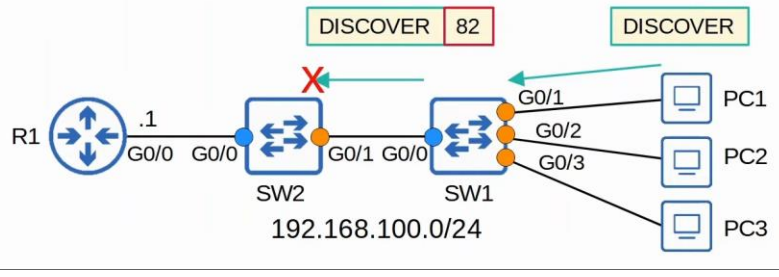
  RELEASE/DECLINE messages: Check if the packet's source IP address and the receiving interface match the entry in the *DHCP Snooping Binding Table*. Match = forward, mismatch = discard

- When a client successfully leases an IP address from a server, create a new entry in the *DHCP Snooping Binding Table*.

DHCP Option 82 (Information Option)

- Option 82, also known as the 'DHCP relay agent information option' is one of many DHCP options.
- It provides additional information about which DHCP relay agent received the client's message, on which interface, in which VLAN, etc.
- DHCP relay agents can add Option 82 to messages they forward to the remote DHCP server.
- With DHCP snooping enabled, by default Cisco switches will add Option 82 to DHCP messages they receive from clients, even if the switch isn't acting as a DHCP relay agent.
- By default, Cisco switches will drop DHCP messages with Option 82 that are received on an untrusted port.

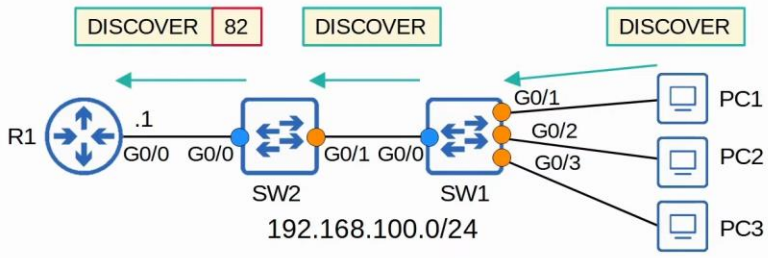
```
SW2#
*Jun  6 01:36:15.298: %DHCP_SNOOPING-5-DHCP_SNOOPING_NONZERO_GIADDR: DHCP_SNOOPING drop message with non-zero giaddr or option82 value on untrusted port, message type: DHCPDISCOVER, MAC sa: 0c29.2f67.e900
```



DHCP Option 82 (Information Option)

- Option 82, also known as the 'DHCP relay agent information option' is one of many DHCP options.
- It provides additional information about which DHCP relay agent received the client's message, on which interface, in which VLAN, etc.
- DHCP relay agents can add Option 82 to messages they forward to the remote DHCP server.
- With DHCP snooping enabled, by default Cisco switches will add Option 82 to DHCP messages they receive from clients, even if the switch isn't acting as a DHCP relay agent.
- By default, Cisco switches will drop DHCP messages with Option 82 that are received on an untrusted port.

```
SW1(config)#no ip dhcp snooping information option
```



Command Review


```
SW1(config)# ip dhcp snooping

SW1(config)# ip dhcp snooping vlan vlan-number

SW1(config)# errdisable recovery cause dhcp-rate-limit

SW1(config)# no ip dhcp snooping information option

SW1(config-if)# ip dhcp snooping trust

SW1(config-if)# ip dhcp snooping limit rate packets-per-second

SW1# show ip dhcp snooping binding
```

The Purpose of DNS

- DNS is used to *resolve* human-readable names (google.com) to IP addresses.
- Machines such as PCs don't use names, they use addresses (ie. IPv4/IPv6).
- Names are much easier for us to use and remember than IP addresses.
  → What's the IP address of youtube.com?
- When you type 'youtube.com' into a web browser, your device will ask a DNS server for the IP address of youtube.com.
- The DNS server(s) your device uses can be manually configured or learned via DHCP.

Wireshark Capture

```
No.   Time              Source          Destination     Protocol  Length  Info
1087  08:55:44.458619   192.168.0.101   8.8.8.8         DNS         71    Standard query 0x0002 A youtube.com
1088  08:55:44.500043   8.8.8.8         192.168.0.101   DNS         87    Standard query response 0x0002 A youtube.com A 172.217.25.110
1089  08:55:44.508888   192.168.0.101   8.8.8.8         DNS         71    Standard query 0x0003 AAAA youtube.com
1115  08:55:44.641775   8.8.8.8         192.168.0.101   DNS         99    Standard query response 0x0003 AAAA youtube.com AAAA 2404:6800:4004:819::200e

> Frame 1087: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{9956EC07-3774-4B11-970D-C8233E7CD172}, id 0
> Ethernet II, Src: Dell_ac:08:67 (78:2b:cb:ac:08:67), Dst: Tp-LinkT_dd:a8:e4 (98:da:c4:dd:a8:e4)
> Internet Protocol Version 4, Src: 192.168.0.101, Dst: 8.8.8.8
> User Datagram Protocol, Src Port: 49286, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x0002
  v Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v youtube.com: type A, class IN
        Name: youtube.com
        [Name Length: 11]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 1088]
```

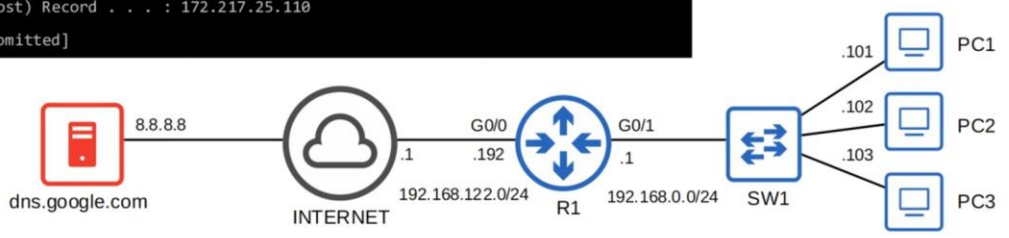
DNS 'A' record = Used to map names to IPv4 addresses.

DNS 'AAAA' record = Used to map names to IPv6 addresses.

DNS Cache

```
C:\Users\user>ipconfig /displaydns

[output omitted]

www.youtube.com
    ----------------------------------------
    Record Name . . . . . : www.youtube.com
    Record Type . . . . . : 5
    Time To Live  . . . . : 98
    Data Length . . . . . : 8
    Section . . . . . . . : Answer
    CNAME Record  . . . . : youtube-ui.l.google.com

[output omitted]

Record Name . . . . . : youtube-ui.l.google.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 98
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 172.217.25.110

[output omitted]
```

Devices will save the DNS server's responses to a local DNS cache.
This means they don't have to query the server every single time they want to access a particular destination.



DNS in Cisco IOS

```
R1(config)#ip dns server

R1(config)#ip host R1 192.168.0.1
R1(config)#ip host PC1 192.168.0.101
R1(config)#ip host PC2 192.168.0.102
R1(config)#ip host PC3 192.168.0.103

R1(config)#ip name-server 8.8.8.8

R1(config)#ip domain lookup
```

Configure R1 to act as a DNS server.

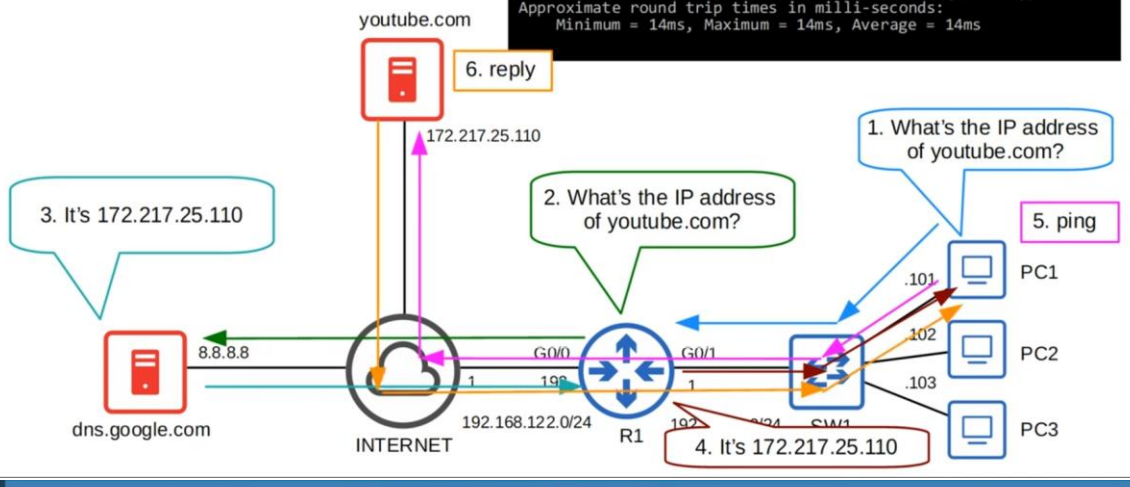
Configure a list of hostname/IP address mappings.

Configure a DNS server that R1 will query if the requested record isn't in its host table.

Enable R1 to perform DNS queries.
(enabled by default)
(old version of the command is **`ip domain-lookup`**)

DNS in Cisco IOS

```
C:\Users\user>ping youtube.com -n 1

Pinging youtube.com [172.217.25.110] with 32 bytes of data:
Reply from 172.217.25.110: bytes=32 time=14ms TTL=117

Ping statistics for 192.168.0.102:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 14ms, Average = 14ms
```



Command Review

Windows:

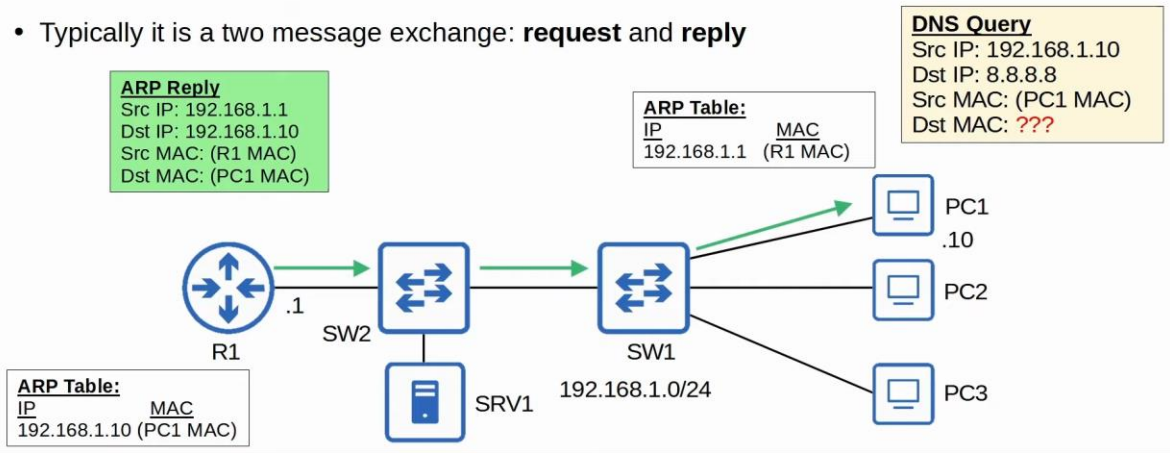
```
C:\Users\user>ipconfig /all

C:\Users\user>nslookup name

C:\Users\user>ipconfig /displaydns

C:\Users\user>ipconfig /flushdns

C:\Users\user>ping ip-address -n number
```

Cisco IOS:

```
R1(config)#ip dns server

R1(config)#ip host hostname ip-address

R1(config)#ip name-server ip-address

R1(config)#ip domain lookup

R1(config)#ip domain name domain-name

R1#show hosts
```

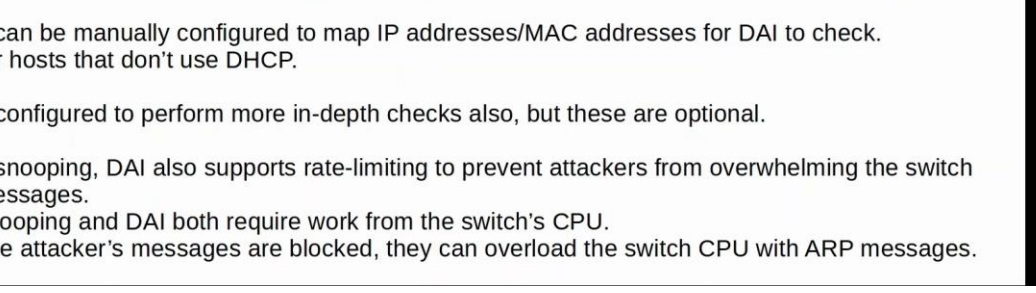
ARP Review

- ARP is used to learn the MAC address of another device with a known IP address.
- For example, a PC will use ARP to learn the MAC address of its default gateway to communicate with external networks.
- Typically it is a two message exchange: **request** and **reply**



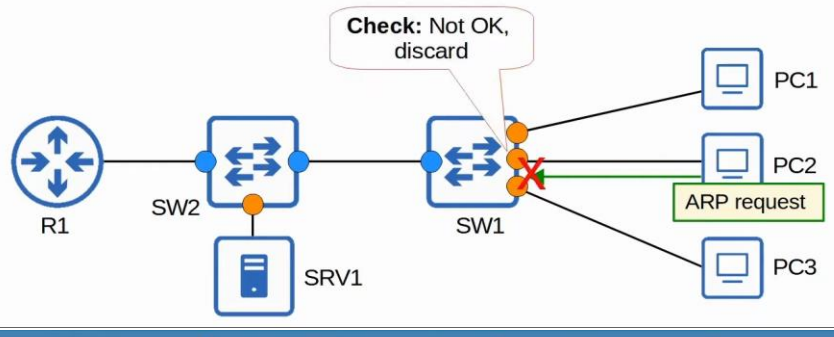
Gratuitous ARP

- A *Gratuitous ARP* message is an ARP reply that is sent without receiving an ARP request.
- It is sent to the broadcast MAC address.
- It allows other devices to learn the MAC address of the sending device without having to send ARP requests.
- Some devices automatically send GARP messages when an interface is enabled, IP address is changed, MAC address is changed, etc.



Dynamic ARP Inspection

- DAI is a security feature of switches that is used to filter ARP messages received on *untrusted* ports.
- DAI only filters ARP messages. Non-ARP messages aren't affected.
- All ports are *untrusted* by default.
  → Typically, all ports connected to other network devices (switches, routers) should be configured as **trusted**, while interfaces connected to end hosts should remain **untrusted**.



Dynamic ARP Inspection Operations

- DAI inspects the sender MAC and sender IP fields of ARP messages received on **untrusted** ports and checks that there is a matching entry in the DHCP snooping binding table.
  → If there is a matching entry, the ARP message is forwarded normally.
  → If there isn't a matching entry, the ARP message is discarded.

```
SW1#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
0C:29:2F:18:79:00   192.168.100.10   86294       dhcp-snooping  1     GigabitEthernet0/3
0C:29:2F:90:91:00   192.168.100.11   86302       dhcp-snooping  1     GigabitEthernet0/1
0C:29:2F:67:E9:00   192.168.100.12   86314       dhcp-snooping  1     GigabitEthernet0/2
Total number of bindings: 3
```

- DAI doesn't inspect messages received on **trusted** ports. They are forwarded as normal.
- **ARP ACLs** can be manually configured to map IP addresses/MAC addresses for DAI to check.
  → Useful for hosts that don't use DHCP.
- DAI can be configured to perform more in-depth checks also, but these are optional.
- Like DHCP snooping, DAI also supports rate-limiting to prevent attackers from overwhelming the switch with ARP messages.
  → DHCP snooping and DAI both require work from the switch's CPU.
  → Even if the attacker's messages are blocked, they can overload the switch CPU with ARP messages.

DAI Optional Checks

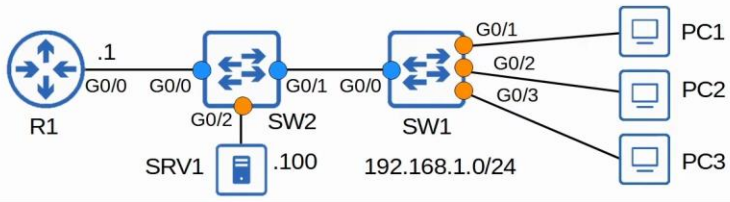
```
SW1(config)#ip arp inspection validate ?
  dst-mac  Validate destination MAC address
  ip       Validate IP addresses
  src-mac  Validate source MAC address
```

dst-mac: Enables validation of the destination MAC address in the Ethernet header against the target MAC address in the ARP body for ARP responses. The device classifies packets with different MAC addresses as invalid and drops them

ip: Enables validation of the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. The device checks the sender IP addresses in all ARP requests and responses and checks the target IP addresses only in ARP responses.

src-mac: Enables validation of the source MAC address in the Ethernet header against the sender MAC address in the ARP body for ARP requests and responses. The devices classifies packets with different MAC addresses as invalid and drops them.

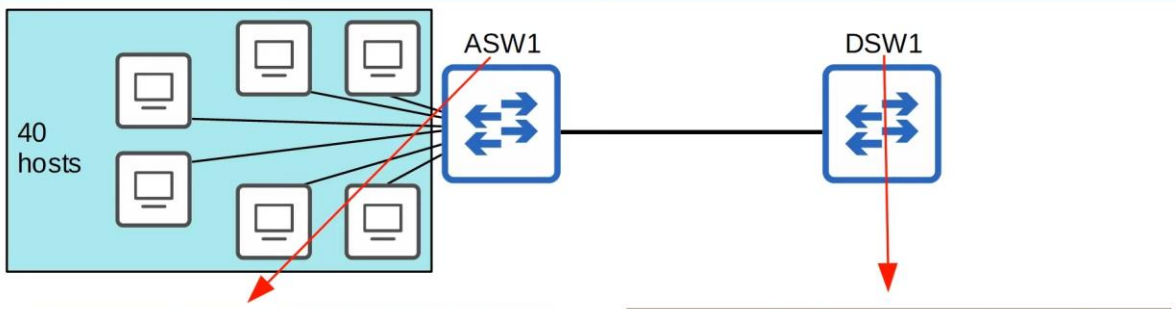
(source: https://www.cisco.com/c/m/en_us/techdoc/dc/reference/cli/n5k/commands/ip-arp-inspection-validate.html)



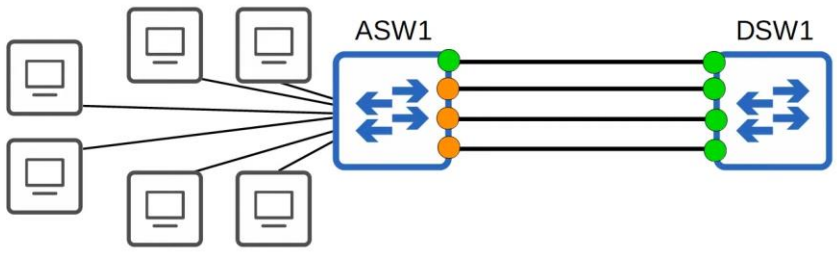
Command Review

```
SW1(config)# ip arp inspection vlan vlan-number

SW1(config)# errdisable recovery cause arp-inspection

SW1(config)# ip arp inspection validate (src-mac | dst-mac | ip)

SW1(config-if)# ip arp inspection trust

SW1(config-if)# ip arp inspection limit rate packets [burst interval seconds]


SW1(config)# arp access-list name

SW1(config-arp-nacl)# permit ip host ip-address mac host mac-address

SW1(config)# ip arp inspection filter arp-acl-name vlan vlan-number


SW1# show ip arp inspection

SW1# show ip arp inspection interfaces
```

EtherChannel

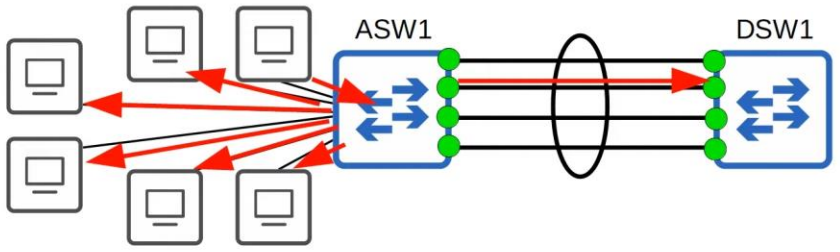


EtherChannel



- If you connect two switches together with multiple links, all except one will be disabled by spanning tree.
- If all of ASW1's interfaces were forwarding, Layer 2 loops would form between ASW1 and DSW1, leading to broadcast storms.
- Other links will be unused unless the active link fails. In that case, one of the inactive links will start forwarding.

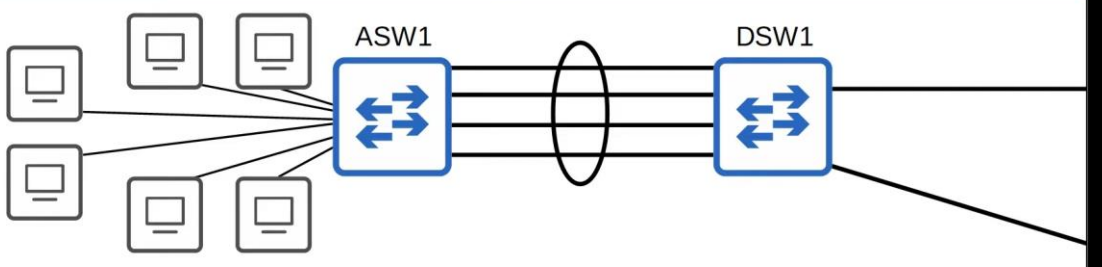
EtherChannel



- EtherChannel groups multiple interfaces together to act as a single interface.
- STP will treat this group as a single interface.

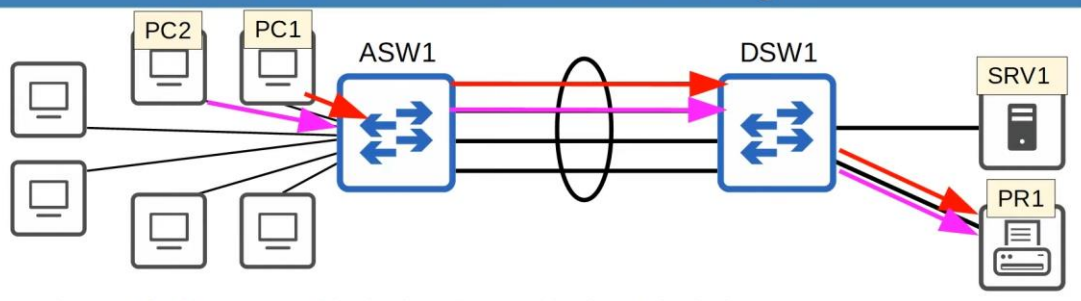
  Traffic using the EtherChannel will be load balanced among the physical interfaces in the group. An algorithm is used to determine which traffic will use which physical interface. More details on this later!

EtherChannel



- EtherChannel groups multiple interfaces together to act as a single interface.
- STP will treat this group as a single interface.
- Some other names for an EtherChannel are:
  Port Channel
  LAG (Link Aggregation Group)

EtherChannel Load-Balancing

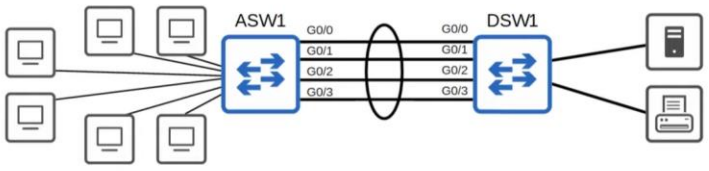


- You can change the inputs used in the interface selection calculation.
- Inputs that can be used:
  Source MAC
  Destination MAC
  Source AND Destination MAC
  Source IP
  Destination IP
  Source AND Destination IP

EtherChannel Configuration

- There are three methods of EtherChannel configuration on Cisco switches:
- PAgP (Port Aggregation Protocol)
  → Cisco proprietary protocol
  → Dynamically negotiates the creation/maintenance of the EtherChannel. (like DTP does for trunks)
- LACP (Link Aggregation Control Protocol)
  → Industry standard protocol (IEEE 802.3ad)
  → Dynamically negotiates the creation/maintenance of the EtherChannel. (like DTP does for trunks)
- Static EtherChannel
  → A protocol isn't used to determine if an EtherChannel should be formed.
  → Interfaces are statically configured to form an EtherChannel.
- Up to 8 interfaces can be formed into a single EtherChannel (LACP allows up to 16, but only 8 will be active, the other 8 will be in standby mode, waiting for an active interface to fail)

EtherChannel Configuration



- Member interfaces must have matching configurations.
  - → Same duplex (full/half)
  - → Same speed
  - → Same switchport mode (access/trunk)
  - → Same allowed VLANs/native VLAN (for trunk interfaces)
- If an interface's configurations do not match the others, it will be excluded from the EtherChannel.

Commands

```
SW(config) port-channel load-balance mode
```
#configures the EtherChannel load-balancing method on the switch

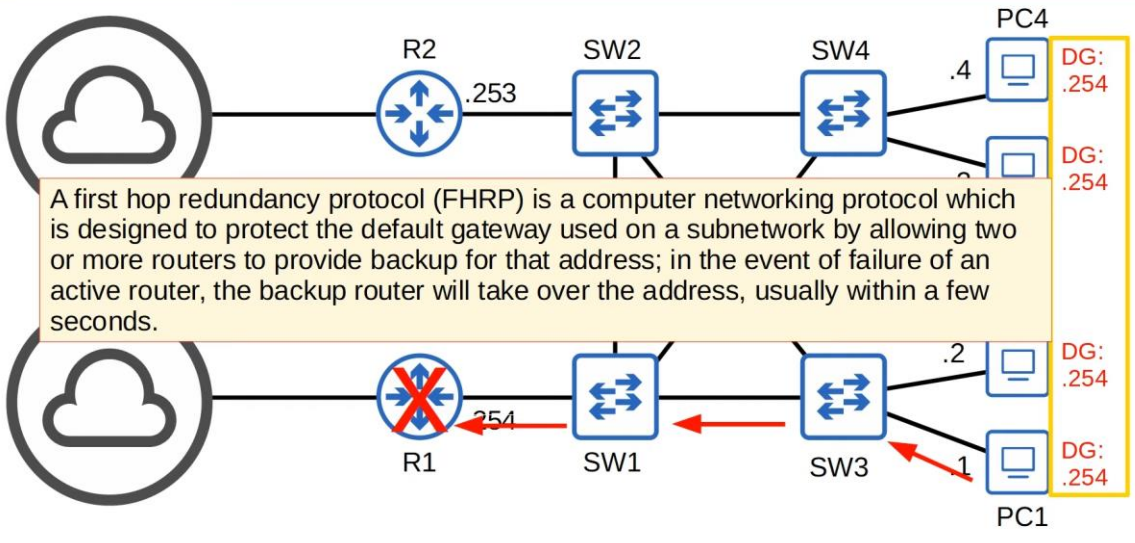
```
SW# show etherchannel load-balance
```
#displays information about the load-balancing settings

```
SW(config-if)# channel-group number mode {desirable|auto|active|passive|on}
```
#configures an interface to be part of an EtherChannel

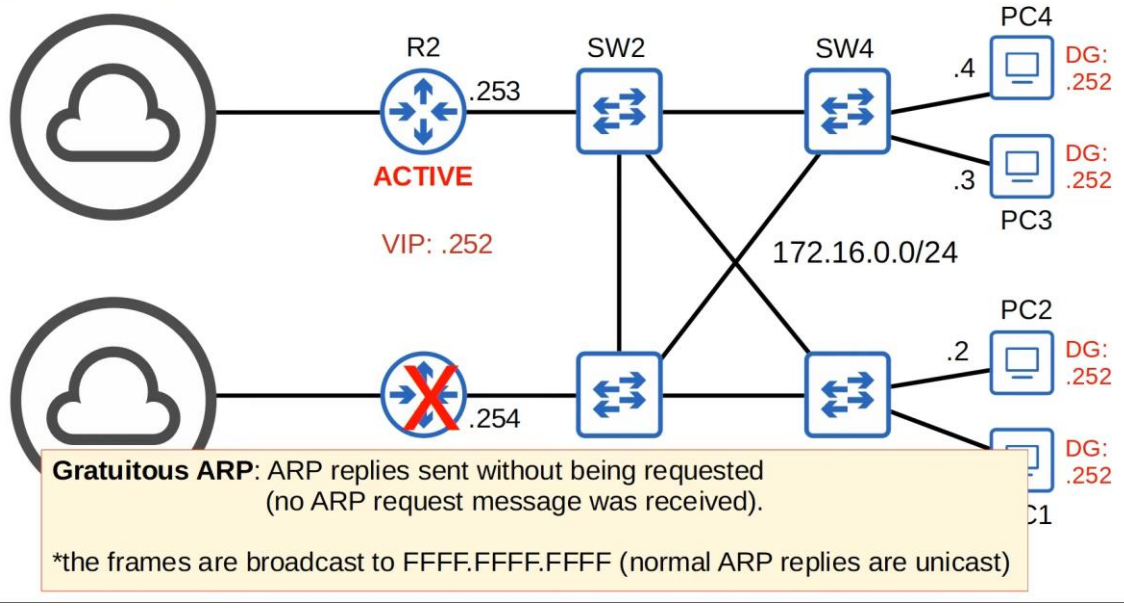
```
SW# show etherchannel summary
```
#displays a summary of EtherChannels on the switch

```
SW# show etherchannel port-channel
```
#displays information about the virtual port-channel interfaces on the switch

First Hop Redundancy Protocols



First Hop Redundancy Protocols



Gratuitous ARP: ARP replies sent without being requested (no ARP request message was received).

*the frames are broadcast to FFFF.FFFF.FFFF (normal ARP replies are unicast)

First Hop Redundancy Protocols

- A **virtual IP** is configured on the two routers, and a **virtual MAC** is generated for the virtual IP (each FHRP uses a different format for the virtual MAC)
- An **active** router and a **standby** router are elected. (different FHRPs use different terms)
- End hosts in the network are configured to use the virtual IP as their default gateway.
- The active router replies to ARP requests using the virtual MAC address, so traffic destined for other networks will be sent to it.
- If the active router fails, the standby becomes the next active router.
  The new active router will send **gratuitous ARP** messages so that switches will update their MAC address tables. It now functions as the default gateway.
- If the old active router comes back online, by default it won't take back its role as the active router. It will become the standby router.
- You can configure 'preemption', so that the old active router does take back its old role.

HSRP (Hot Standby Router Protocol)

- Cisco proprietary.
- An **active** and **standby** router are elected.
- There are two versions: **version 1** and **version 2**.
  Version 2 adds IPv6 support and increases the number of *groups* that can be configured.
- Multicast IPv4 address: v1 = 224.0.0.2
  v2 = 224.0.0.102
- Virtual MAC address: v1 = 0000.0c07.ac**XX** (**XX** = HSRP group number)
  v2 = 0000.0c9f.f**XXX** (**XXX** = HSRP group number)

```
0000.0c07.ac01
```

```
0000.0c9f.f001
```

- In a situation with multiple subnets/VLANs, you can configure a different active router in each subnet/VLAN to load balance.

GLBP (Gateway Load Balancing Protocol)

- Cisco proprietary
- Load balances among multiple routers within a single subnet
- An **AVG** (**Active Virtual Gateway)** is elected.
- Up to four **AVFs (Active Virtual Forwarders)** are assigned by the AVG (the AVG itself can be an AVF, too)
- Each AVF acts as the default gateway for a portion of the hosts in the subnet.
- Multicast IPv4 address: 224.0.0.102
- Virtual MAC address: 0007.b400.**XXYY** (**XX** = GLBP group number, **YY** = AVF number)

```
0007.b400.0101
```

Comparing FHRPs

FHRP	Terminology	Multicast IP	Virtual MAC	Cisco proprietary?
HSRP	Active/Standby	v1: 224.0.0.2 v2: 224.0.0.102	v1: 0000.0c07.acXX v2: 0000.0c9f.fXXX	Yes
VRRP	Master/Backup	224.0.0.18	0000.5e00.01XX	No
GLBP	AVG / AVF	224.0.0.102	0007.b400.XXYY	Yes

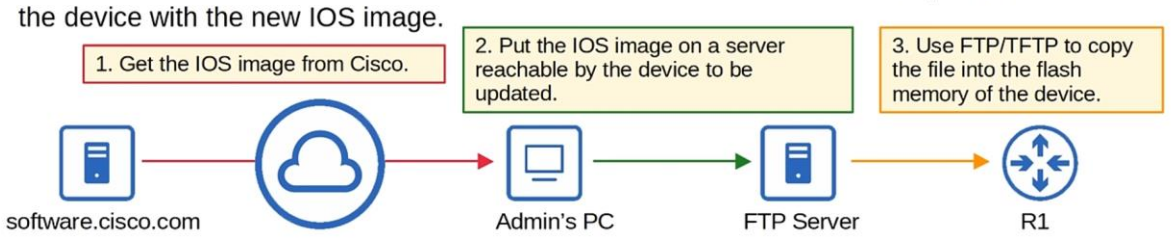
Things we covered

- The purpose of FHRPs
- HSRP (Hot Standby Router Protocol)
- VRRP (Virtual Router Redundancy Protocol)
- GLBP (Gateway Load Balancing Protocol)
- Basic HSRP Configuration

```
R1(config-if)# standby version 2
R1(config-if)# standby group-number ip virtual-ip
R1(config-if)# standby group-number priority priority
```

FTP & TFTP

- FTP (File Transfer Protocol) and TFTP (Trivial File Transfer Protocol) are industry standard protocols used to transfer files over a network.
- They both use a client-server model.
  - → Clients can use FTP or TFTP to copy files from a server.
  - → Clients can use FTP or TFTP to copy files to a server.
- As a network engineer, the most common use for FTP/TFTP is in the process of upgrading the operating system of a network device.
- You can use FTP/TFTP to download the newer version of IOS from a server, and then reboot the device with the new IOS image.

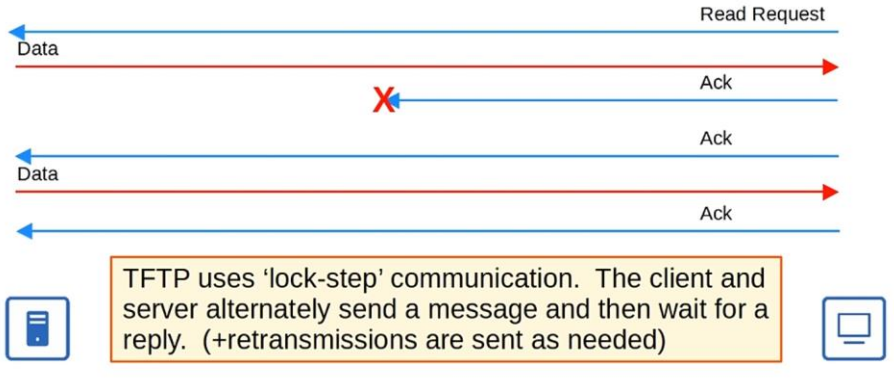


Trivial File Transfer Protocol

- TFTP was first standardized in 1981.
- Named 'Trivial' because it is simple and has only basic features compared to FTP.
  - → Only allows a client to copy a file to or from a server.
- Was released after FTP, but is not a replacement for FTP. It is another tool to use when lightweight simplicity is more important than functionality.
- No authentication (username/PW), so servers will respond to all TFTP requests.
- No encryption, so all data is sent in plain text.
- Best used in a controlled environment to transfer small files quickly.
- TFTP servers listen on **UDP port 69**.
- UDP is connectionless and doesn't provide reliability with retransmissions.
- However, TFTP has similar built-in features within the protocol itself.

TFTP Reliability

- Every TFTP data message is acknowledged.
  - → If the client is transferring a file to the server, the server will send Ack messages.
  - → If the server is transferring a file to the client, the client will send Ack messages.
- Timers are used, and if an expected message isn't received in time, the waiting device will re-send its previous message.



TFTP 'Connections'

- TFTP file transfers have three phases:
  1: **Connection**: TFTP client sends a request to the server, and the server responds back, initializing the connection.

  2: **Data Transfer**: The client and server exchange TFTP messages. One sends data and the other sends acknowledgments.

  3: **Connection Termination**: After the last data message has been sent, a final acknowledgment is sent to terminate the connection.

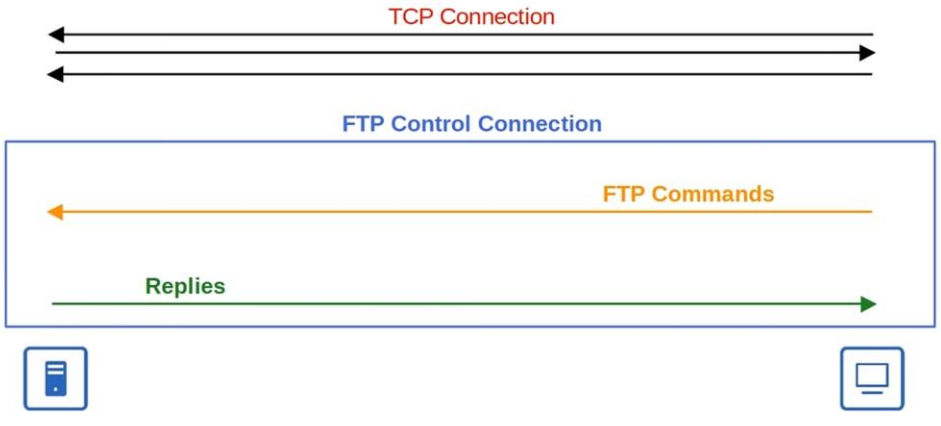
File Transfer Protocol

- FTP was first standardized in 1971.
- FTP uses **TCP ports 20 and 21**.
- Usernames and passwords are used for authentication, however there is no encryption.
- For greater security, FTPS (FTP over SSL/TLS) can be used. ← Upgrade to FTP
- SSH File Transfer Protocol (SFTP) can also be used for greater security. ← New protocol
- FTP is more complex than TFTP and allows not only file transfers, but clients can also navigate file directories, add and remove directories, list files, etc.
- The client sends *FTP commands* to the server to perform these functions.

https://en.wikipedia.org/wiki/List_of_FTP_commands

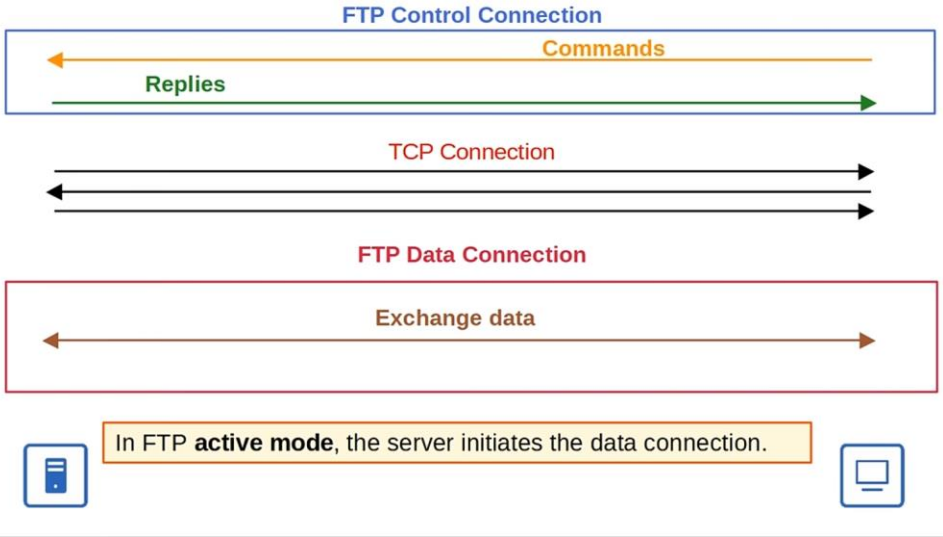
FTP Control Connections

- FTP uses two types of connections:
  → An **FTP control** connection (**TCP 21**) is established and used to send FTP commands and replies.
  → When files or data are to be transferred, separate **FTP data** (**TCP 20**) connections are established and terminated as needed.



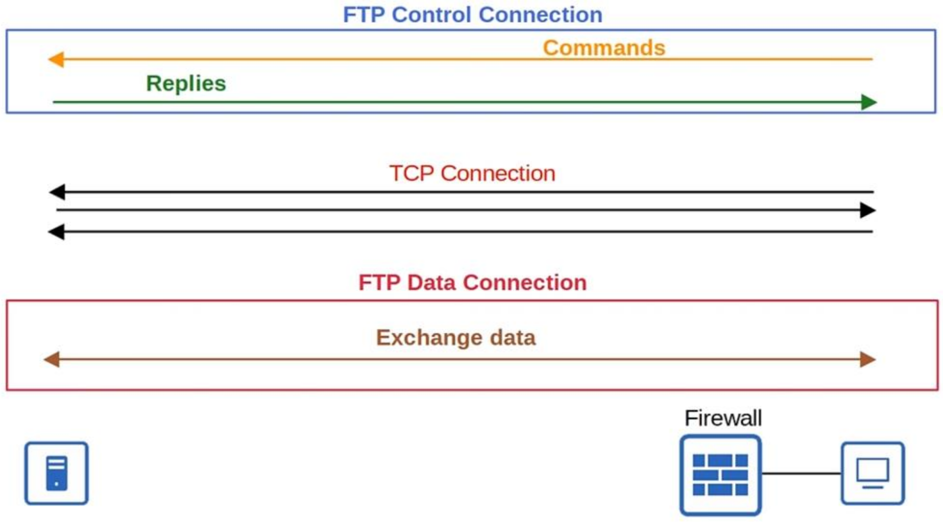
Active Mode FTP Data Connections

- The default method of establishing FTP data connections is **active mode**, in which the server initiates the TCP connection.



Passive Mode FTP Data Connections

- In FTP **passive mode**, the client initiates the data connection. This is often necessary when the client is behind a firewall, which could block the incoming connection from the server.



Copying Files (TFTP)

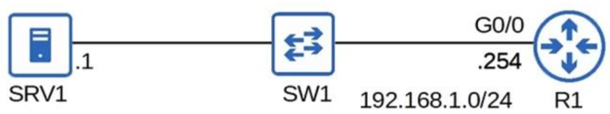
`copy source destination`

```
R1#copy tftp: flash:
Address or name of remote host []? 192.168.1.1
Source filename []? c2900-universalk9-mz.SPA.155-3.M4a.bin
Destination filename [c2900-universalk9-mz.SPA.155-3.M4a.bin]?

Accessing tftp://192.168.1.1/c2900-universalk9-mz.SPA.155-3.M4a.bin....
Loading c2900-universalk9-mz.SPA.155-3.M4a.bin from
192.168.1.1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 33591768 bytes]

33591768 bytes copied in 4.01 secs (879550 bytes/sec)
```

Enter the TFTP server IP.

Enter the file name on the server

Enter the name you want to save it as on flash (hit enter to accept the default)



Command Review

```
R1# show file systems
R1# show version
R1# show flash
R1# copy source destination
R1(config)# boot system filepath
R1(config)# ip ftp username username
R1(config)# ip ftp password password
```

Private IPv4 Addresses (RFC 1918)

- IPv4 doesn't provide enough addresses for all devices that need an IP address in the modern world.
- The long-term solution is to switch to IPv6.
- There are three main short-term solutions:
  1) CIDR
  2) Private IPv4 addresses
  3) NAT
- RFC 1918 specifies the following IPv4 address ranges as private:

  10.0.0.0/8 (10.0.0.0 to 10.255.255.255) → Class A

  172.16.0.0/12 (172.16.0.0 to 172.31.255.255) → Class B

  192.168.0.0/16 (192.168.0.0 to 192.168.255.255) → Class C

Private IPv4 Addresses (RFC 1918)

- RFC 1918 specifies the following IPv4 address ranges as private:
  10.0.0.0/8 (10.0.0.0 to 10.255.255.255)
  172.16.0.0/12 (172.16.0.0 to 172.31.255.255)
  192.168.0.0/16 (192.168.0.0 to 192.168.255.255)

- You are free to use these addresses in your networks. They don't have to be globally unique.

```
C:\Users\user>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.0.167
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

*Private IP addresses cannot be used over the Internet!

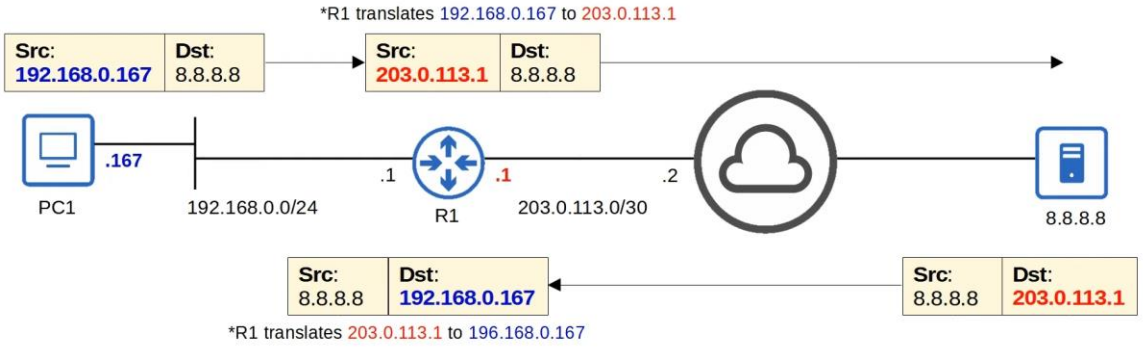
- Two problems:
  1) Duplicate addresses
  2) Private IP addresses can't be used over the Internet, so the PCs can't access the Internet.

Network Address Translation (NAT)

- Network Address Translation (NAT) is used to modify the source and/or destination IP addresses of packets.

- There are various reasons to use NAT , but the most common reason is to allow hosts with private IP addresses to communicate with other hosts over the Internet.

- For the CCNA you have to understand **source NAT** and how to configure it on Cisco routers.

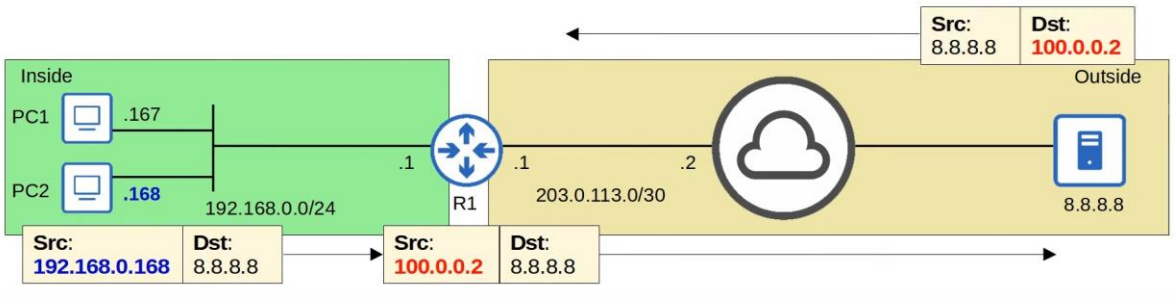


Static NAT

- **Static NAT** involves statically configuring one-to-one mappings of private IP addresses to public IP addresses.

- An *inside local* IP address is mapped to an *inside global* IP address.
  → **Inside Local** = The IP address of the *inside* host, from the perspective of the local network
  *the IP address actually configured on the inside host, usually a private address

  → **Inside Global** = The IP address of the *inside* host, from the perspective of *outside* hosts
  *the IP address of the inside host after NAT, usually a public address



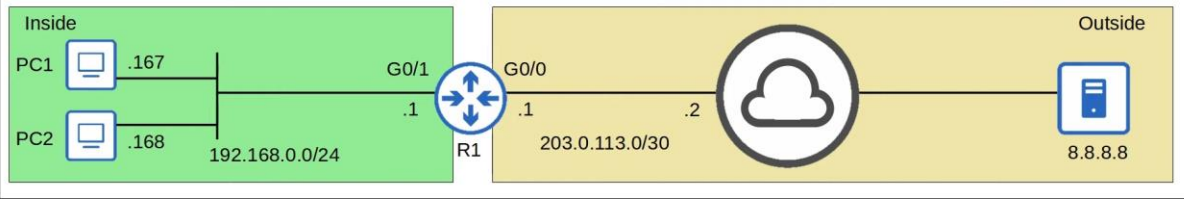
Static NAT Configuration

```
R1(config)#int g0/1
R1(config-if)#ip nat inside
```
Define the 'inside' interface(s) connected to the internal network.

```
R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```
Define the 'outside' interface(s) connected to the external network.

```
R1(config)#ip nat inside source static 192.168.0.167 100.0.0.1
R1(config)#ip nat inside source static 192.168.0.168 100.0.0.2
R1(config)#exit
```
Configure the one-to-one IP address mappings.
ip nat inside source static *inside-local-ip inside-global-ip*

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 100.0.0.1:56310    192.168.0.167:56310 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                ---
udp 100.0.0.2:62321    192.168.0.168:62321 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.2          192.168.0.168      ---                ---
```



show ip nat translations

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 100.0.0.1:56310    192.168.0.167:56310 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                 ---
udp 100.0.0.2:62321    192.168.0.168:62321 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.2          192.168.0.168      ---                 ---
```

Unless **destination NAT** is used, these two addresses will be the same.

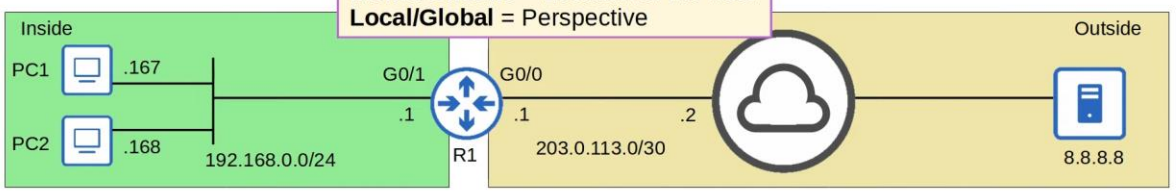
→ **Inside Local** = The IP address of the *inside* host, from the perspective of the local network
*the IP address actually configured on the inside host, usually a private address

→ **Inside Global** = The IP address of the *inside* host, from the perspective of *outside* hosts
*the IP address of the inside host after NAT, usually a public address

→ **Outside Local** = The IP address of the *outside* host, from the perspective of the local network

→ **Outside Global** = The IP address of the *outside* host, from the perspective of the outside network

Inside/Outside = Location of the host
Local/Global = Perspective



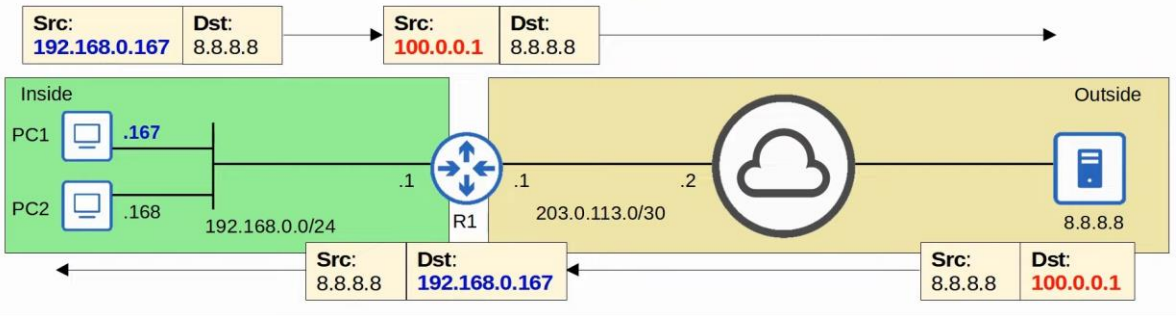
Command Review

```
R1(config-if)# ip nat inside
R1(config-if)# ip nat outside
R1(config)# ip nat inside source static inside-local-ip inside-global-ip
R1# show ip nat translations
R1# show ip nat statistics
R1# clear ip nat translation *
```

Static NAT

- **Static NAT** involves statically configuring one-to-one mappings of private IP addresses to public IP addresses.
- When traffic from the internal host is sent to the outside network, the router will translate the source address.

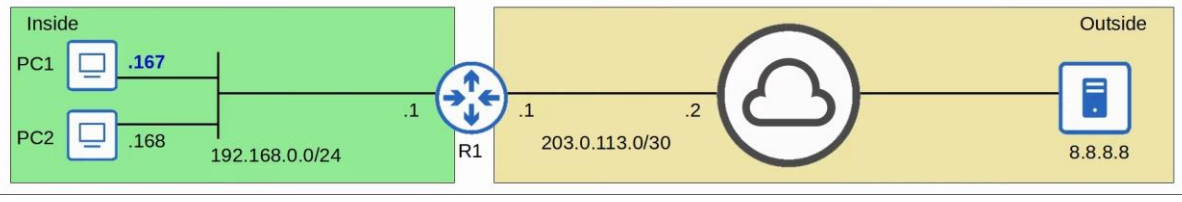
Static NAT: **192.168.0.167** = **100.0.0.1**
192.168.0.168 = **100.0.0.2**



Dynamic NAT

- In **dynamic NAT**, the router dynamically maps *inside local* addresses to *inside global* addresses as needed.
- An ACL is used to identify which traffic should be translated.
  → If the source IP is **permitted** by the ACL, the source IP will be translated.
  → If the source IP is **denied** by the ACL, the source IP will NOT be translated. *the traffic will NOT be dropped!
- A NAT pool is used to define the available *inside global* addresses.

On R1:
ACL 1: permit **192.168.0.0/24**
deny any
POOL1: **100.0.0.1** to **100.0.0.10**
If a packet with a source IP permitted by **ACL 1** arrives, translate the source IP to an address from **POOL1**.



Dynamic NAT

- In **dynamic NAT**, the router dynamically maps *inside local* addresses to *inside global* addresses as needed.
- An ACL is used to identify which traffic should be translated.
  → If the source IP is **permitted** by the ACL, the source IP will be translated.
  → If the source IP is **denied** by the ACL, the source IP will NOT be translated. *the traffic will NOT be dropped!
- A NAT pool is used to define the available *inside global* addresses that can be used.
- Although they are dynamically assigned, the mappings are still one-to-one (one *inside local* IP address per *inside global* IP address).
- If there aren't enough *inside global* IP addresses available (=all are currently being used), it is called 'NAT pool exhaustion'.
  → If a packet from another inside host arrives and needs NAT but there are no available addresses, the router will drop the packet.
  → The host will be unable to access outside networks until one of the *inside global* IP addresses becomes available.
  → Dynamic NAT entries will time out automatically if not used, or you can clear them manually.

NAT Pool Exhaustion

Source IP	Translated Source IP
192.168.0.167	100.0.0.1
192.168.0.168	100.0.0.2
192.168.0.100	100.0.0.3
192.168.0.12	100.0.0.4
192.168.0.28	100.0.0.5
192.168.0.56	100.0.0.6
192.168.0.202	100.0.0.7
192.168.0.221	100.0.0.8
192.168.0.116	100.0.0.9
192.168.0.188	100.0.0.10
192.168.0.98	**No address available! Router will drop the packet**

Dynamic NAT Configuration

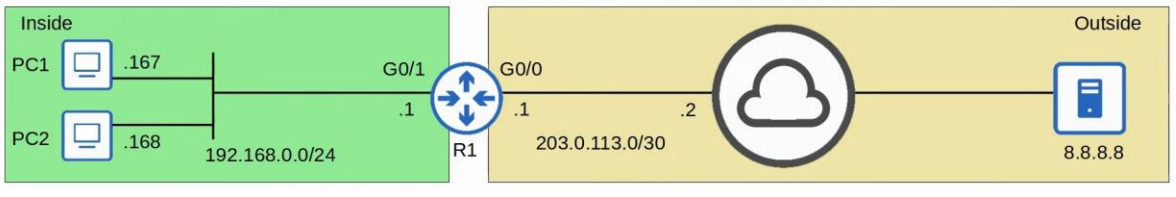
```
R1(config)#int g0/1
R1(config-if)#ip nat inside
```
Define the 'inside' interface(s) connected to the internal network.

```
R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```
Define the 'outside' interface(s) connected to the external network.

```
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```
Define the traffic that should be translated.
*Traffic permitted by this ACL will be translated.

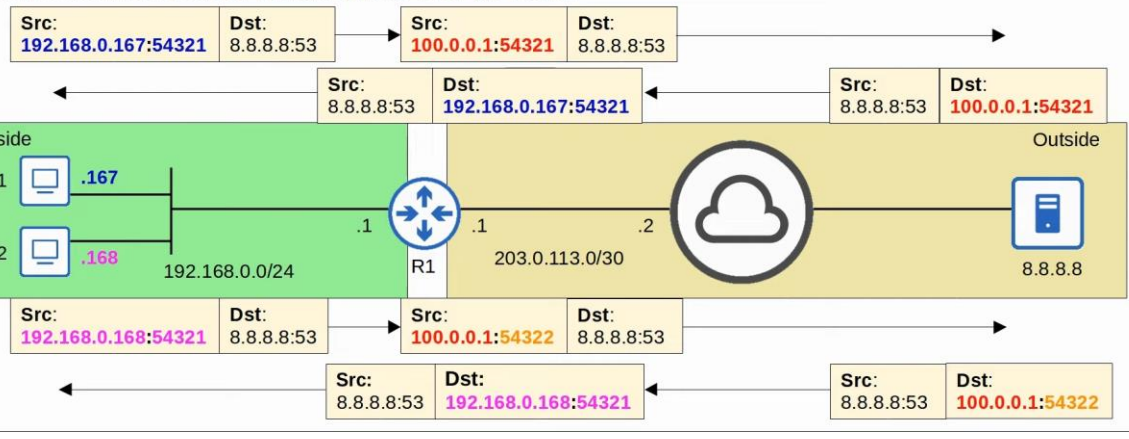
```
R1(config)#ip nat pool POOL1 100.0.0.0 100.0.0.255 prefix-length 24
```
Define the pool of inside global IP addresses.
*instead of **`prefix-length 24`**, you can use **`netmask 255.255.255.0`**

```
R1(config)#ip nat inside source list 1 pool POOL1
```
Configure dynamic NAT by mapping the ACL to the pool.



PAT (NAT Overload)

- **PAT** (aka **NAT overload**) translates both the IP address and the port number (if necessary).
- By using a unique port number for each communication flow, a single public IP address can be used by many different internal hosts. (port number are 16 bits = over 65,000 available port numbers).
- The router will keep track of which *inside local* address is using which *inside global* address and port.
- Because many inside hosts can share a single public IP, PAT is very useful for preserving public IP addresses, and it is used in networks all over the world.



PAT Configuration (pool)

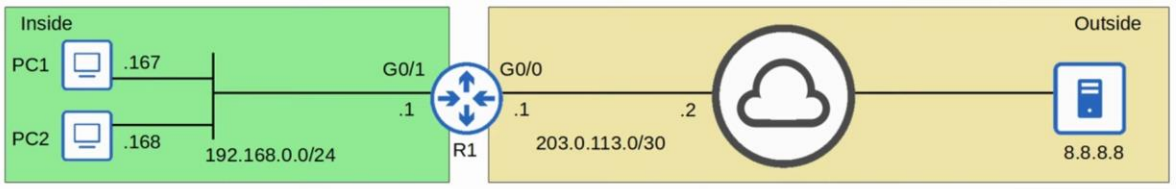
```
R1(config)#int g0/1
R1(config-if)#ip nat inside
```
Define the 'inside' interface(s) connected to the internal network.

```
R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```
Define the 'outside' interface(s) connected to the external network.

```
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```
Define the traffic that should be translated.
*Traffic permitted by this ACL will be translated.

```
R1(config)#ip nat pool POOL1 100.0.0.0 100.0.0.3 prefix-length 24
```
Define the pool of inside global IP addresses.

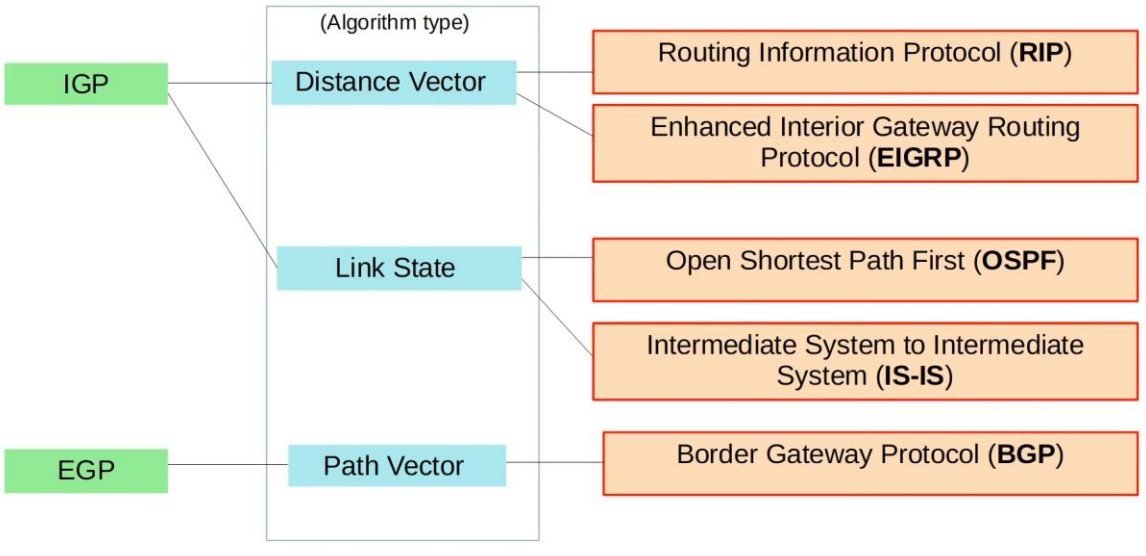
```
R1(config)#ip nat inside source list 1 pool POOL1 overload
```
Configure PAT by mapping the ACL to the pool and using the **overload** keyword at the end.



Command Review

```
R1(config)# ip nat pool pool-name start-ip end-ip prefix-length prefix-length
R1(config)# ip nat pool pool-name start-ip end-ip netmask subnet-mask
R1(config)# ip nat inside source list access-list pool pool-name
R1(config)# ip nat inside source list access-list pool pool-name overload
R1(config)# ip nat inside source list access-list interface interface overload
```

Types of Dynamic Routing Protocols



OSPF

- Stands for **Open Shortest Path First**
- Uses the **Shortest Path First** algorithm of Dutch computer scientist Edsger Dijkstra. (aka **Dijkstra's algorithm** ← remember that name!)
- Three versions:
  OSPFv1 (1989): OLD, not in use anymore
  OSPFv2 (1998): Used for IPv4
  OSPFv3 (2008): Used for IPv6 (can also be used for IPv4, but usually v2 is used)
- Routers store information about the network in LSAs (Link State Advertisements), which are organized in a structure called the LSDB (Link State Database).
- Routers will **flood** LSAs until all routers in the OSPF *area* develop the same map of the network (LSDB).

OSPF

- In OSPF, there are three main steps in the process of sharing LSAs and determining the best route to each destination in the network.

1) **Become neighbors** with other routers connected to the same segment.

2) **Exchange LSAs** with neighbor routers.

3) **Calculate the best routes** to each destination, and insert them into the routing table.

OSPF Areas

- OSPF uses **areas** to divide up the network.
- Small networks can be *single-area* without any negative effects on performance.
- In larger networks, a single-area design can have negative effects:
  -the SPF algorithm takes more time to calculate routes
  -the SPF algorithm requires exponentially more processing power on the routers
  -the larger LSDB takes up more memory on the routers
  -any small change in the network causes every router to flood LSAs and run the SPF algorithm again
- By dividing a large OSPF network into several smaller areas, you can avoid the above negative effects.

OSPF Areas

- An **area** is a set of routers and links that share the same LSDB.
- The **backbone area** (area 0) is an area that all other areas must connect to.
- Routers with all interfaces in the same area are called **internal routers**.
- Routers with interfaces in multiple areas are called **area border routers** (**ABRs**).
- Routers connected to the backbone area (area 0) are called **backbone routers**.
- An **intra-area route** is a route to a destination inside the same OSPF area.
- An **interarea route** is a route to a destination in a different OSPF area.

show ip protocols


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

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 172.16.1.14
  It is an autonomous system boundary router
 Redistributing External Routes from,
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.12.0 0.0.0.3 area 0
    10.0.13.0 0.0.0.3 area 0
    172.16.1.0 0.0.0.15 area 0
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    4.4.4.4              110      00:00:08
    2.2.2.2              110      00:01:07
    3.3.3.3              110      00:01:07
    192.168.4.254        110      00:02:29
  Distance: (default is 110)
```

Router ID order of priority:
1) Manual configuration
2) Highest IP address on a loopback interface
3) Highest IP address on a physical interface

```
R1(config-router)#router-id ?
  A.B.C.D  OSPF router-id in IP address format

R1(config-router)#router-id 1.1.1.1
% OSPF: Reload or use "clear ip ospf process" command, for this to take effect
```

show ip protocols

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

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
  It is an autonomous system boundary router
 Redistributing External Routes from,
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.12.0 0.0.0.3 area 0
    10.0.13.0 0.0.0.3 area 0
    172.16.1.0 0.0.0.15 area 0
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    2.2.2.2              110      00:01:40
    3.3.3.3              110      00:01:40
    4.4.4.4              110      00:01:40
  Distance: (default is 110)
```

- An **autonomous system boundary router** (ASBR) is an OSPF router that connects the OSPF network to an external network.
- R1 is connected to the Internet. By using the **default-information originate** command, R1 becomes an ASBR.

```
R1(config-router)#maximum-paths ?
  <1-32>  Number of paths

R1(config-router)#maximum-paths 8
```

OSPF Cost

- OSPF's metric is called **cost**
- It is automatically calculated based on the bandwidth (speed) of the interface.
- It is calculated by dividing a **reference bandwidth** value by the interface's bandwidth.
- The default reference bandwidth is 100 mbps.
  - **Reference**: 100 mbps / **Interface**: 10 mbps = cost of 10
  - **Reference**: 100 mbps / **Interface**: 100 mbps = cost of 1
  - **Reference**: 100 mbps / **Interface**: 1000 mbps = cost of 1??
  - **Reference**: 100 mbps / **Interface**: 10000 mbps = cost of 1??
- All values less than 1 will be converted to 1.
- Therefore FastEthernet, Gigabit Ethernet, 10Gig Ethernet, etc. are equal and all have a cost of 1 by default.
- You can (and should!) change the reference bandwidth with this command:
  `R1(config-router)# auto-cost reference-bandwidth` *megabits-per-second*

OSPF Cost

- One more option to change the OSPF cost of an interface is to change the bandwidth of the interface with the **bandwidth** command.
- The formula to calculate OSPF cost is **reference bandwidth** / **interface bandwidth**
- Although the bandwidth matches the interface speed by default, changing the interface bandwidth doesn't actually change the speed at which the interface operates.
- The bandwidth is just a value that is used to calculate OSPF cost, EIGRP metric, etc.
- To change the speed at which the interface operates, use the **speed** command.
- Because the bandwidth value is used in other calculations, it is not recommended to change this value to alter the interface's OSPF cost.
- It is recommended that you change the reference bandwidth, and then use the **ip ospf cost** command to change the cost of individual interfaces if you want.

```
R1(config-if)#bandwidth ?
  <1-10000000>   Bandwidth in kilobits
  inherit        Specify how bandwidth is inherited
  qos-reference  Reference bandwidth for QOS test
  receive        Specify receive-side bandwidth
```

OSPF Cost

- Three ways to modify the OSPF cost:

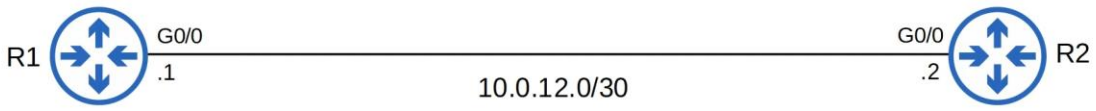
1) Change the **reference bandwidth**:
`R1(config-router)# auto-cost reference-bandwidth` *megabits-per-second*

2) Manual configuration
`R1(config-if)# ip ospf cost` *cost*

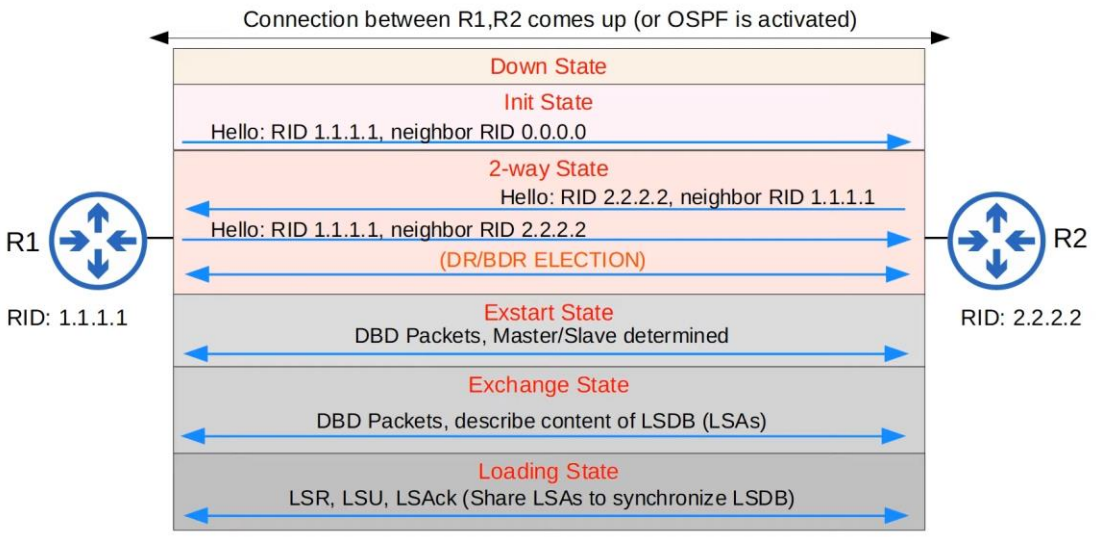
3) Change the **interface bandwidth**
`R1(config-if)# bandwidth` *kilobits-per-second*

OSPF Neighbors

- Making sure that routers successfully become OSPF neighbors is the main task in configuring and troubleshooting OSPF.
- Once routers become neighbors, they automatically do the work of sharing network information, calculating routes, etc.
- When OSPF is activated on an interface, the router starts sending OSPF **hello** messages out of the interface at regular intervals (determined by the **hello timer**). These are used to introduce the router to potential OSPF neighbors.
- The default hello timer is 10 seconds on an Ethernet connection.
- Hello messages are multicast to 224.0.0.5 (multicast address for all OSPF routers)
- OSPF messages are encapsulated in an IP header, with a value of 89 in the Protocol field.



OSPF Neighbors



OSPF

- In OSPF, there are three main steps in the process of sharing LSAs and determining the best route to each destination in the network.

1) **Become neighbors** with other routers connected to the same segment.

2) **Exchange LSAs** with neighbor routers.

3) **Calculate the best routes** to each destination, and insert them into the routing table.

OSPF

Type	Name	Purpose
1	**Hello**	Neighbor discovery and maintenance.
2	**Database Description (DBD)**	Summary of the LSDB of the router. Used to check if the LSDB of each router is the same.
3	**Link-State Request (LSR)**	Requests specific LSAs from the neighbor.
4	**Link-State Update (LSU)**	Sends specific LSAs to the neighbor.
5	**Link-State Acknowledgement (LSAck)**	Used to acknowledge that the router received a message.

OSPF metric (cost)

- **Reference bandwidth** / **interface bandwidth** = cost (values less than 1 are converted to 1)
- Default reference bandwidth = 100 mbps
- Modify the reference bandwidth:
  `R1(config-router)# auto-cost reference-bandwidth megabits-per-second`
- Manually configure the cost of an interface:
  `R1(config-if)# ip ospf cost cost`
- Modify the interface bandwidth:
  `R1(config-if)# bandwidth kilobits-per-second`
- Total cost of *outgoing* interfaces = metric of the route

OSPF Network Types

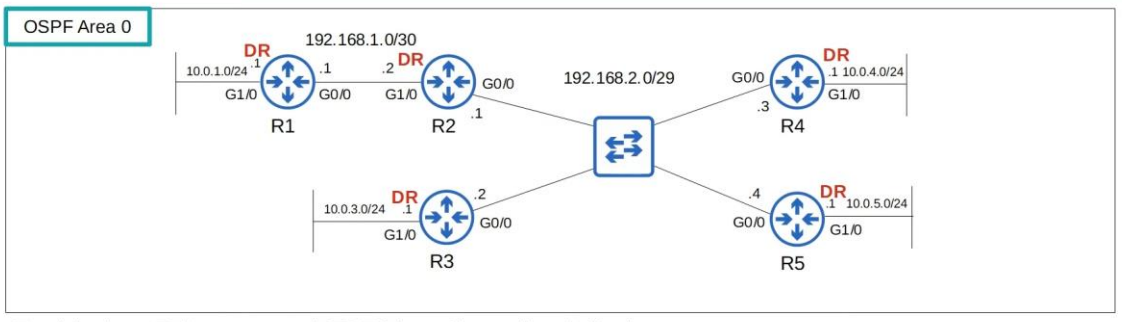
- The OSPF 'network type' refers to the type of connection between OSPF neighbors (Ethernet, etc)
- There are three main OSPF network types:

  Broadcast
  -enabled by default on **Ethernet** and **FDDI** (Fiber Distributed Data Interfaces) interfaces

  Point-to-point
  -enabled by default on **PPP** (Point-to-Point Protocol) and **HDLC** (High-Level Data Link Control) interfaces

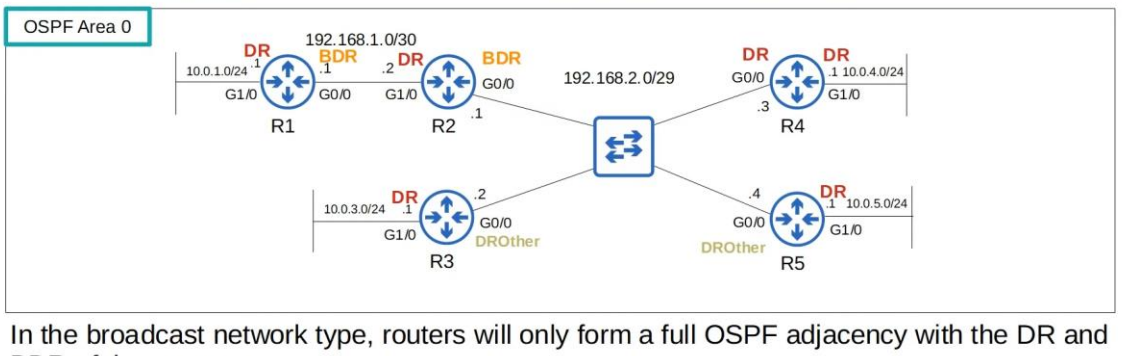
  Non-broadcast
  -enabled by default on **Frame Relay** and **X.25** interfaces

OSPF Broadcast Network Type



- Enabled on **Ethernet** and **FDDI** interfaces by default.
- Routers *dynamically discover* neighbors by sending/listening for OSPF Hello messages using multicast address 224.0.0.5.
- A **DR** (designated router) and **BDR** (backup designated router) must be elected on each subnet (only DR if there are no OSPF neighbors, ie. R1's G1/0 interface)
- Routers which aren't the DR or BDR become a **DROther**.

OSPF Broadcast Network Type



- In the broadcast network type, routers will only form a full OSPF adjacency with the DR and BDR of the segment.
- Therefore, routers only exchange LSAs with the DR and BDR. DROthers will not exchange LSAs with each other.
- All routers will still have the same LSDB, but this reduces the amount of LSAs flooding the network.

Serial Interfaces

- The default encapsulation is HDLC.
- You can configure PPP encapsulation with this command:
  `R1(config-if)# encapsulation ppp`
- One side is DCE, one side is DTE.
- Identify which side is DCE/DTE:
  `R1# show controllers interface-id`
- You must configure the clock rate on the DCE side:
  `R1(config-if)# clock rate bits-per-second`

Configure the OSPF Network Type

Broadcast	Point-to-point
Default on **Ethernet**, **FDDI** interfaces	Default on **HDLC**, **PPP** (serial) interfaces
DR/DBR elected	No DR/BDR
Neighbors dynamically discovered	Neighbors dynamically discovered
Default timers: Hello 10, Dead 40	Default timers: Hello 10, Dead 40

OSPF Neighbor Requirements

1) Area number must match

2) Interfaces must be in the same subnet

3) OSPF process must not be **`shutdown`**

4) OSPF Router IDs must be unique

5) Hello and Dead timers must match

6) Authentication settings must match

7) IP MTU settings must match

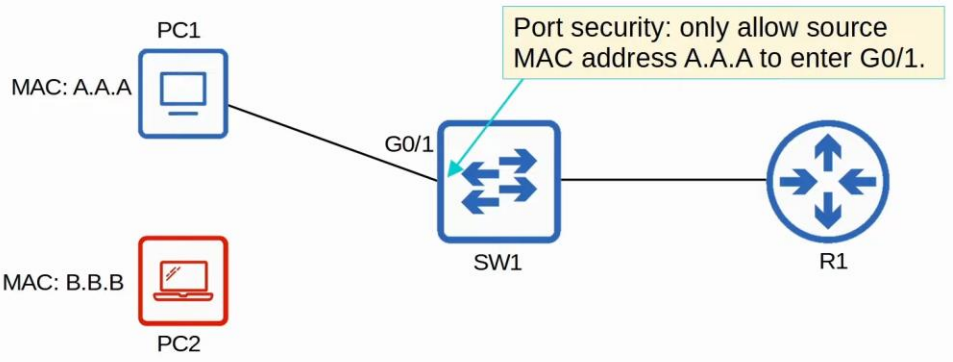
Can become OSPF neighbors, but OSPF doesn't operate properly.

OSPF LSA Types

- **Type 1 (Router LSA)**
  -Every OSPF router generates this type of LSA.
  -It identifies the router using its router ID.
  -It also lists networks attached to the router's OSPF-activated interfaces.
- **Type 2 (Network LSA)**
  -Generated by the DR of each 'multi-access' network (ie. the **broadcast** network type).
  -Lists the routers which are attached to the multi-access network.
- **Type 5 (AS-External LSA)**
  -Generated by ASBRs to describe routes to destinations outside of the AS (OSPF domain).

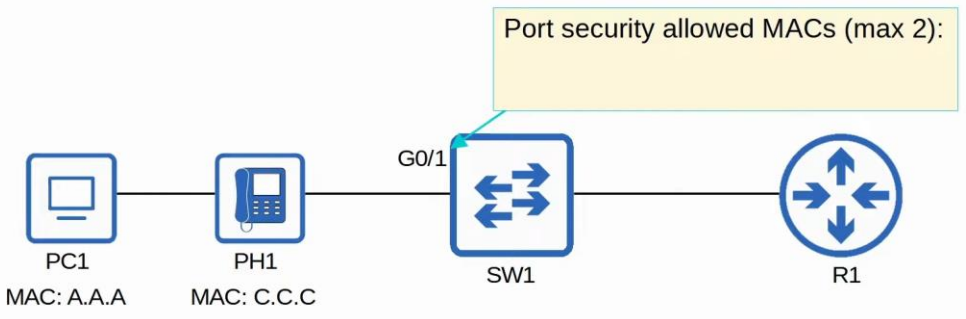
Port Security

- Port security is a security feature of Cisco switches.
- It allows you to control which source MAC address(es) are allowed to enter the switchport.
- If an unauthorized source MAC address enters the port, an action will be taken.
  → The default action is to place the interface in an 'err-disabled' state.



Port Security

- When you enable port security on an interface with the default settings, one MAC address is allowed.
  → You can configure the allowed MAC address manually.
  → If you don't configure it manually, the switch will allow the first source MAC address that enters the interface.
- You can change the maximum number of MAC addresses allowed.



Why port security?

- Port security allows network admins to control which devices are allowed to access the network.
- However, MAC address spoofing is a simple task.
  → It's easy to configure a device to send frames with a different source MAC address.
- Rather than manually specifying the MAC addresses allowed on each port, port security's ability to limit the number of MAC addresses allowed on an interface is more useful.
- Think of the DHCP starvation attack carried out in the Day 48 Lab video.
  → the attacker spoofed thousands of fake MAC addresses
  → the DHCP server assigned IP addresses to these fake MAC addresses, exhausting the DHCP pool
  → the switch's MAC address table can also become full due to such an attack
- Limiting the number of MAC addresses on an interface can protect against those attacks.

Violation Modes

There are three different violation modes that determine what the switch will do if an unauthorized frame enters an interface configured with port security.

- **Shutdown**
  → Effectively shuts down the port by placing it in an err-disabled state.
  → Generates a Syslog and/or SNMP message when the interface is disabled.
  → The violation counter is set to 1 when the interface is disabled.
- **Restrict**
  → The switch discards traffic from unauthorized MAC addresses.
  → The interface is NOT disabled.
  → Generates a Syslog and/or SNMP message each time an unauthorized MAC is detected.
  → The violation counter is incremented by 1 for each unauthorized frame.
- **Protect**
  → The switch discards traffic from unauthorized MAC addresses.
  → The interface is NOT disabled.
  → It does NOT generate Syslog/SNMP messages for unauthorized traffic.

Secure MAC address aging

```
SW1#show port-security interface g0/1
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   : 000a.000a.000a:1
Security Violation Count   : 0
```

- By default secure MAC addresses will not 'age out' (`Aging Time : 0 mins`)
  → Can be configured with **`switchport port-security aging time`** *minutes*
- The default aging type is **Absolute**
  → **Absolute**: After the secure MAC address is learned, the aging timer starts and the MAC is removed after the timer expires, even if the switch continues receiving frames from that source MAC address.
  → **Inactivity**: After the secure MAC address is learned, the aging timer starts but is reset every time a frame from that source MAC address is received on the interface.
  → Aging type is configured with **`switchport port-security aging type`** {**`absolute`** | **`inactivity`**}
- Secure Static MAC aging (addresses configured with **`switchport port-security mac-address`** *x.x.x*) is disabled by default.

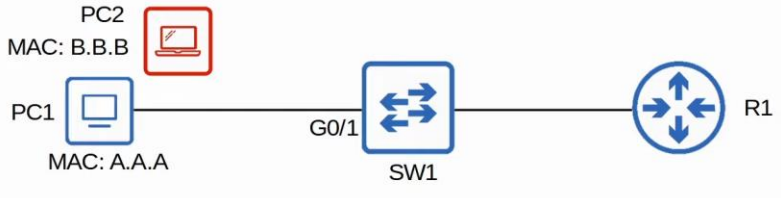
Sticky Secure MAC Addresses

- 'Sticky' secure MAC address learning can be enabled with the following command:
  `SW1(config-if)#` **`switchport port-security mac-address sticky`**
- When enabled, dynamically-learned secure MAC addresses will be added to the running config like this:
  `switchport port-security mac-address sticky` *mac-address*
- The 'sticky' secure MAC addresses will never age out.
  → You need to save the running-config to the startup-config to make them truly permanent (or else they will not be kept if the switch restarts)
- When you issue the **`switchport port-security mac-address sticky`** command, all current dynamically-learned secure MAC addresses will be converted to sticky secure MAC addresses.
- If you issue the **`no switchport port-security mac-address sticky`** command, all current sticky secure MAC addresses will be converted to regular dynamically-learned secure MAC addresses.

MAC Address Table

- Secure MAC addresses will be added to the MAC address table like any other MAC address.
  → Sticky and Static secure MAC addresses will have a type of STATIC
  → Dynamically-learned secure MAC addresses will have a type of DYNAMIC
  → You can view all secure MAC addresess with **show mac address-table secure**

```
SW1#show mac address-table secure
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    000a.000a.000a    STATIC      Gi0/1
Total Mac Addresses for this criterion: 1
```

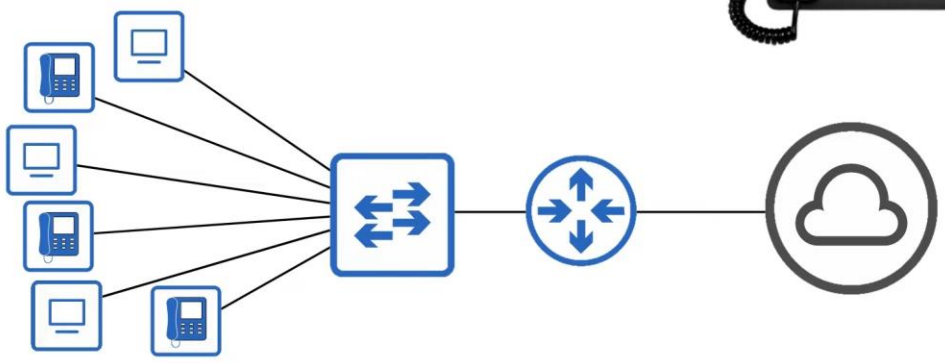


Command Review

```
SW1# show mac address-table secure
SW1# show port-security
SW1# show port-security interface interface
SW1# show errdisable recovery
SW1(config)# errdisable recovery cause psecure-violation
SW1(config)# errdisable recovery interval seconds
SW1(config-if)# switchport port-security
SW1(config-if)# switchport port-security mac-address mac-address
SW1(config-if)# switchport port-security mac-address sticky
SW1(config-if)# switchport port-security violation {shutdown | restrict | protect}
SW1(config-if)# switchport port-security aging time minutes
SW1(config-if)# switchport port-security aging type {absolute | inactivity}
SW1(config-if)# switchport port-security aging static
```

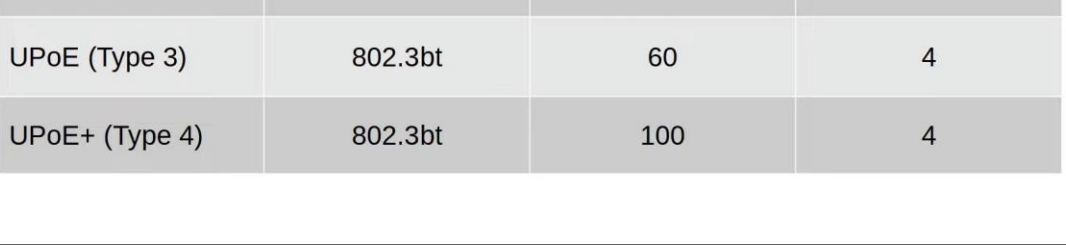
IP Phones

- Traditional phones operate over the *public switched telephone network* (PSTN).
- Sometimes this is called POTS (Plain Old Telephone Service).
- IP phones use VoIP (Voice over IP) technologies to enable phone calls over an IP network, such as the Internet.
- IP phones are connected to a switch just like any other end host.



IP Phones

- IP phones have an internal 3-port switch.
  - → 1 port is the 'uplink' to the external switch.
  - → 1 port is the 'downlink' to the PC.
  - → 1 port connects internally to the phone itself.
- This allows the PC and the IP phone to share a single switch port. Traffic from the PC passes through the IP phone to the switch.
- It is recommended to separate 'voice' traffic (from the IP phone) and 'data' traffic (from the PC) by placing them in separate VLANs.
  - → This can be accomplished using a *voice VLAN*
  - → Traffic from the PC will be untagged, but traffic from the phone will be tagged with a VLAN ID



Power over Ethernet (PoE)

- Too much electrical current can damage electrical devices.
- PoE has a process to determine if a connected device needs power, and how much power it needs.

  → When a device is connected to a PoE-enabled port, the PSE (switch) sends low power signals, monitors the response, and determines how much power the PD needs.

  → If the device needs power, the PSE supplies the power to allow the PD to boot.

  → The PSE continues to monitor the PD and supply the required amount of power (but not too much!)
- *Power policing* can be configured to prevent a PD from taking too much power.

  → **`power inline police`** configures power policing with the default settings: disable the port and send a Syslog message if a PD draws too much power.
    - → equivalent to **`power inline police action err-disable`**
    - → the interface will be put in an 'error-disabled' state and can be re-enabled with **`shutdown`** followed by **`no shutdown`**.

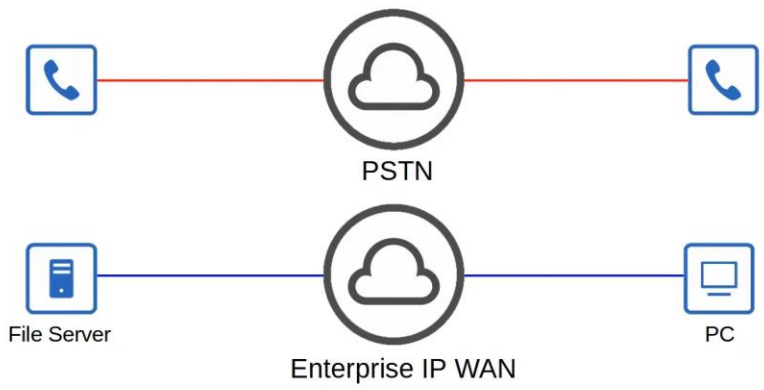
  → **`power inline police action log`** does not shut down the interface if the PD draws too much power. It will restart the interface and send a Syslog message.

Power over Ethernet (PoE)

Name	Standard #	Watts	Powered Wire Pairs
Cisco Inline Power (ILP)	Made by Cisco, not standard	7	2
PoE (Type 1)	802.3af	15	2
PoE+ (Type 2)	802.3at	30	2
UPoE (Type 3)	802.3bt	60	4
UPoE+ (Type 4)	802.3bt	100	4

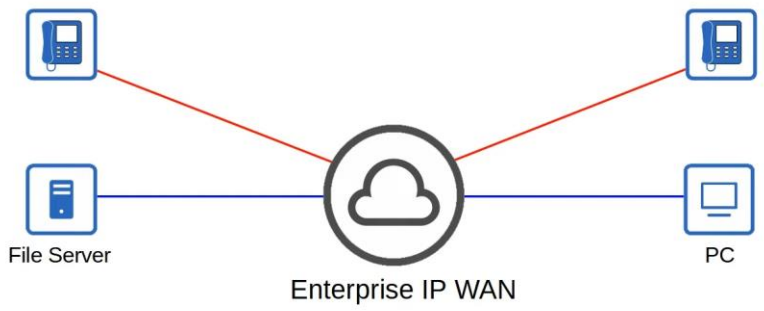
Quality of Service (QoS)

- Voice traffic and data traffic used to use entirely separate networks.
  → Voice traffic used the PSTN
  → Data traffic used the IP network (enterprise WAN, Internet, etc)
- QoS wasn't necessary as the different kinds of traffic didn't compete for bandwidth.



Quality of Service (QoS)

- Modern networks are typically *converged networks* in which IP phones, video traffic, regular data traffic, etc all share the same IP network.
- This enable cost savings as well as more advanced features for voice and video traffic, for example integrations with collaboration software (Cisco WebEx, Microsoft Teams, etc).
- However, the different kinds of traffic now have to compete for bandwidth.
- QoS is a set of tools used by network devices to apply different treatment to different packets.



Quality of Service (QoS)

- QoS is used to manage the following characteristics of network traffic:
  1) **Bandwidth**
  → The overall capacity of the link, measured in bits per second (Kbps, Mbps, Gbps, etc)
  → QoS tools allow you to reserve a certain amount of a link's bandwidth for specific kinds of traffic. For example: 20% voice traffic, 30% for specific kinds of data traffic, leaving 50% for all other traffic.

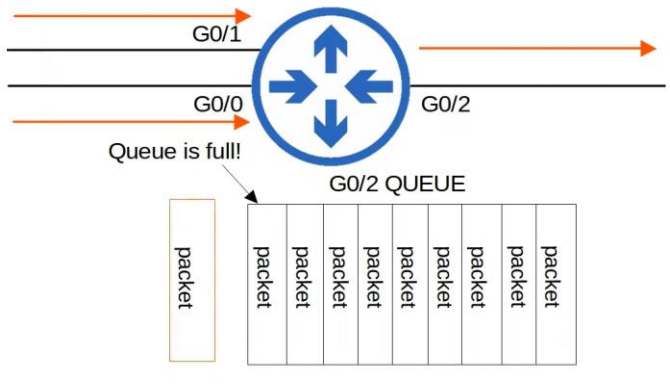
  2) **Delay**
  → The amount of time it takes traffic to go from source to destination = **one-way delay**
  → The amount of time it takes traffic to go from source to destination and return = **two-way delay**

  3) **Jitter**
  → The variation in one-way delay between packets sent by the same application
  → IP phones have a 'jitter buffer' to provide a fixed delay to audio packets.

  4) **Loss**
  → The % of packets sent that do not reach their destination
  → Can be caused by faulty cables.
  → Can also be caused when a device's packet *queues* get full and the device starts discarding packets.

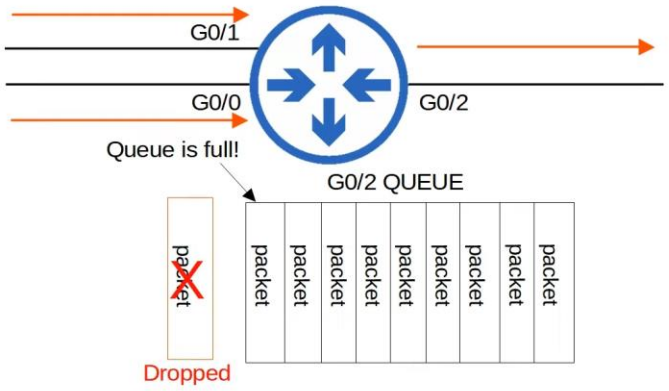
QoS - Queuing

- If a network device receives messages faster than it can forward them out of the appropriate interface, the messages are placed in a queue.
- By default, queued messages will be forwarded in a First In First Out (FIFO) manner.
  → Messages will be sent in the order they are received.
- If the queue is full new packets will be dropped.



QoS - Queuing

- If a network device receives messages faster than it can forward them out of the appropriate interface, the messages are placed in a queue.
- By default, queued messages will be forwarded in a First In First Out (FIFO) manner.
  → Messages will be sent in the order they are received.
- If the queue is full new packets will be dropped.
- This is called *tail drop*.



QoS - Queuing

- **Tail drop** is harmful because it can lead to **TCP global synchronization**.

- Review of the **TCP sliding window**:
  → Hosts using TCP use the 'sliding window' increase/decrease the rate at which they send traffic as needed.
  → When a packet is dropped it will be re-transmitted.
  → When a drop occurs, the sender will reduce the rate it sends traffic.
  → It will then gradually increase the rate again.

- When the queue fills up and **tail drop** occurs, all TCP hosts sending traffic will slow down the rate at which they send traffic.

- They will all then increase the rate at which they send traffic, which rapidly leads to more congestion, dropped packets, and the process repeats again.

Network congestion → Tail drop → Global TCP window size decrease → Network underutilized → Global TCP window size increase

QoS - Queuing

- A solution to prevent tail drop and TCP global synchronization is **Random Early Detection** (RED).

- When the amount of traffic in the queue reaches a certain threshold, the device will start randomly dropping packets from select TCP flows.

- Those TCP flows that dropped packets will reduce the rate at which traffic is sent, but you will avoid global TCP synchronization, in which ALL TCP flows reduce and then increase the rate of transmission at the same time in waves.

- In standard RED, all kinds of traffic are treated the same.

- An improved version, **Weighted Random Early Detection** (WRED), allows you to control which packets are dropped depending on the traffic class.

- We will cover traffic classes and details about how QoS actually works in the next video.

Classification

- The purpose of QoS is to give certain kinds of network traffic priority over others during congestion.

- **Classification** organizes network traffic (packets) into traffic classes (categories).

- Classification is fundamental to QoS. To give priority to certain types of traffic, you have to identify which types of traffic to give priority to.

- There are many methods of classifying traffic. Some examples:
  → An ACL. Traffic which is permitted by the ACL will be given certain treatment, other traffic will not.
  → **NBAR** (Network Based Application Recognition) performs a *deep packet inspection*, looking beyond the Layer 3 and Layer 4 information up to Layer 7 to identify the specific kind of traffic.
  → In the Layer 2 and Layer 3 headers there are specific fields used for this purpose.

- The **PCP** (Priority Code Point) field of the 802.1Q tag (in the Ethernet header) can be used to identify high/low priority traffic.
  → Only when there is a dot1q tag!

- The **DSCP** (Differentiated Services Code Point) field of the IP header can also be used to identify high/low priority traffic.

PCP/CoS

Ethernet Header

Preamble	SFD	Destination	Source	802.1Q	Type/ Length

802.1Q tag format

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

- PCP is also known as CoS (Class of Service). Its use is defined by IEEE 802.1p.
- 3 bits = 8 possible values (2^3 = 8).

'Best effort' delivery means there is no guarantee that data is delivered or that it meets any QoS standard. This is regular traffic, not high-priority.

IP phones **mark** call signaling traffic (used to establish calls) as PCP3.
They **mark** the actual voice traffic as PCP5.

PCP value	Traffic types
0	Best effort (default)
1	Background
2	Excellent effort
3	Critical applications
4	Video
5	Voice
6	Internetwork control
7	Network control

DSCP

DSCP (Differentiated Services Code Point)	ECN

6 bits = 64 values (0-63)

- RFC 2474 (1998) defines the DSCP field, and other 'DiffServ' RFCs elaborate on its use.
- With IPP updated to DSCP, new standard markings had to be decided upon.
  → By having generally agreed upon standard markings for different kinds of traffic, QoS design & implementation is simplified, QoS works better between ISPs and enterprises, among other benefits.
- You should be aware of the following standard markings:
  → Default Forwarding (DF) – best effort traffic
  → Expedited Forwarding (EF) – low loss/latency/jitter traffic (usually voice)
  → Assured Forwarding (AF) – A set of 12 standard values
  → Class Selector (CS) – A set of 8 standard values, provides backward compatibility with IPP

DF / EF

DF (Default Forwarding):

32	16	8	4	2	1		
0	0	0	0	0	0		

- **DF** is used for best-effort traffic.
- The DSCP marking for DF is 0.

EF (Expedited Forwarding):

32	16	8	4	2	1		
1	0	1	1	1	0		

- **EF** is used for traffic that requires low loss/latency/jitter.
- The DSCP marking for EF is 46.

AF

- **AF** (Assured Forwarding) defines four traffic classes. All packets in a class have the same priority.
- Within each class, there are three levels of *drop precedence*.
  → Higher drop precedence = more likely to drop the packet during congestion

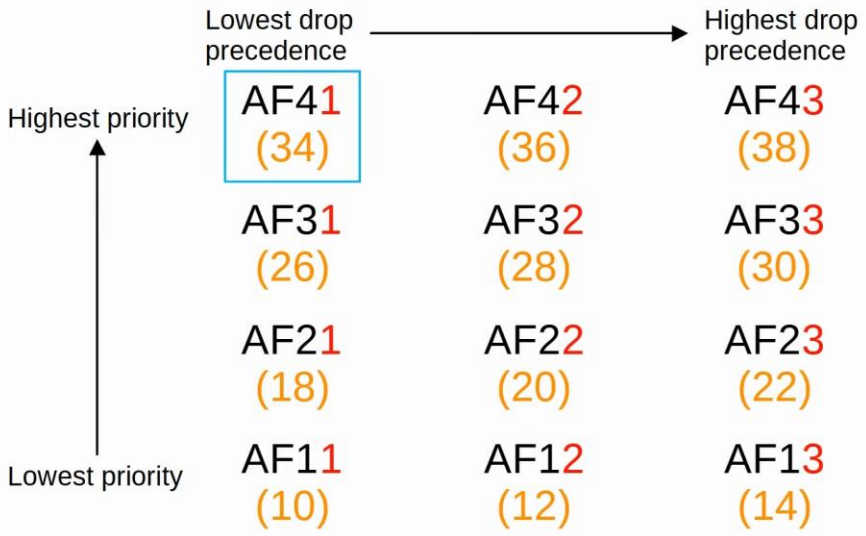
(32)	(16)	(8)	(4)	(2)	(1)		
4	2	1	2	1			
1	0	0	1	1	0		
	X		Y				

= AF43

(DSCP 38)

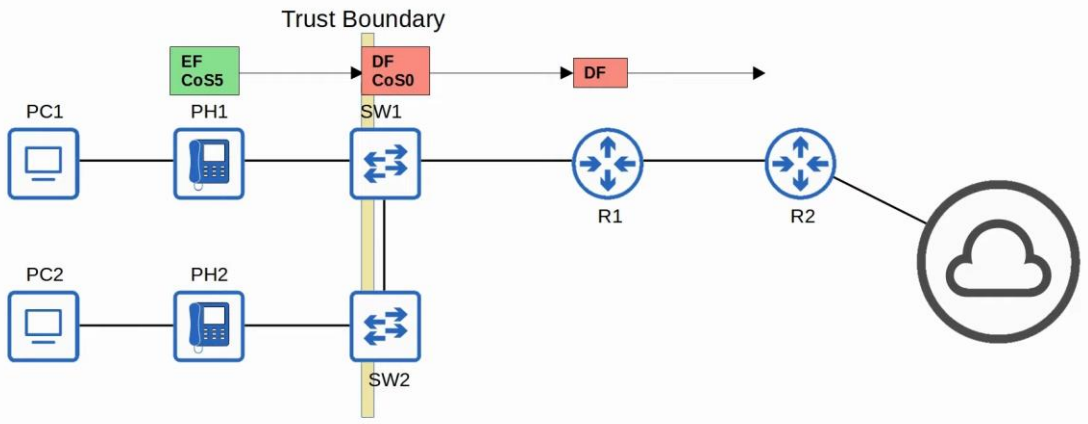
Formula to convert from AF value to decimal DSCP value: 8X + 2Y

AF



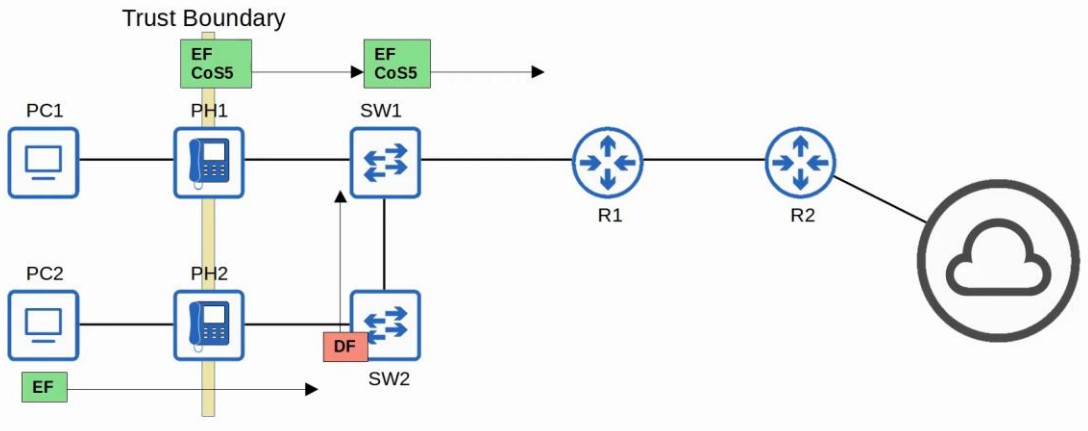
Trust Boundaries

- The *trust boundary* of a network defines where devices trust/don't trust the QoS markings of received messages.
- If the markings are trusted, the device will forward the message without changing the markings.
- If the markings aren't trusted, the device will change the markings according to the configured policy.



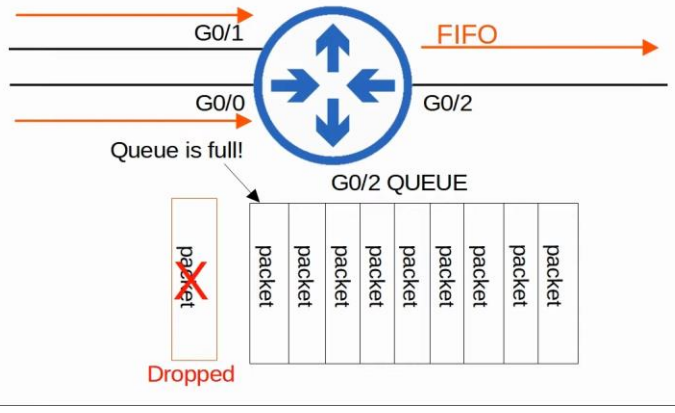
Trust Boundaries

- If an IP phone is connected to the switch port, it is recommended to move the trust boundary to the IP phones.
- This is done via configuration on the switch port connected to the IP phone.
- If a user marks their PC's traffic with a high priority, the marking will be changed (not trusted)



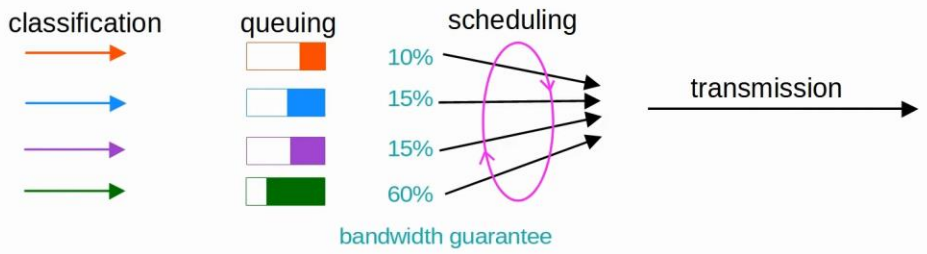
Queuing/Congestion Management

- When a network device receives traffic at a faster rate than it can forward the traffic out of the appropriate interface, packets are placed in that interface's queue as they wait to be forwarded.
- When the queue becomes full, packets that don't fit in the queue are dropped (tail drop).
- RED and WRED drop packets early to avoid tail drop.



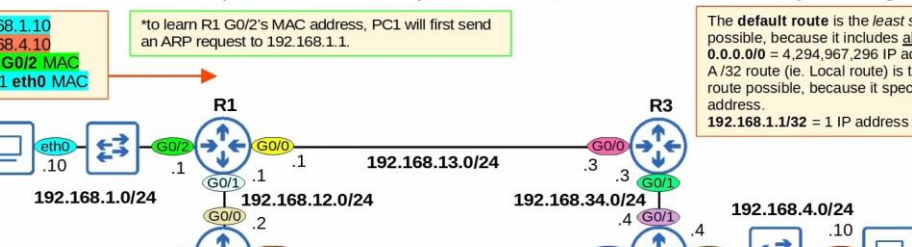
Queuing/Congestion Management

- A common scheduling method is *weighted round-robin*.
  → **round-robin** = packets are taken from each queue in order, cyclically
  → **weighted** = more data is taken from high priority queues each time the scheduler reaches that queue
- **CBWFQ** (Class-Based Weighted Fair Queuing) is a popular method of scheduling, using a weighted round-robin scheduler while guaranteeing each queue a certain percentage of the interface's bandwidth during congestion.
- Round-robin scheduling is not ideal for voice/video traffic. Even if the voice/video traffic receives a guaranteed minimum amount of bandwidth, round-robin can add delay and jitter because even the high priority queues have to wait their turn in the scheduler.



Queuing/Congestion Management

- **LLQ** (Low Latency Queuing) designates one (or more) queues as *strict priority queues*.
  → This means that if there is traffic in the queue, the scheduler will always take the next packet from that queue until it is empty.
- This is very effective for reducing the delay and jitter of voice/video traffic.
- However, it has the downside of potentially starving other queues if there is always traffic in the designated strict priority queue.
  → *Policing* (next slide) can control the amount of traffic allowed in the strict priority queue so that it can't take all of the link's bandwidth.

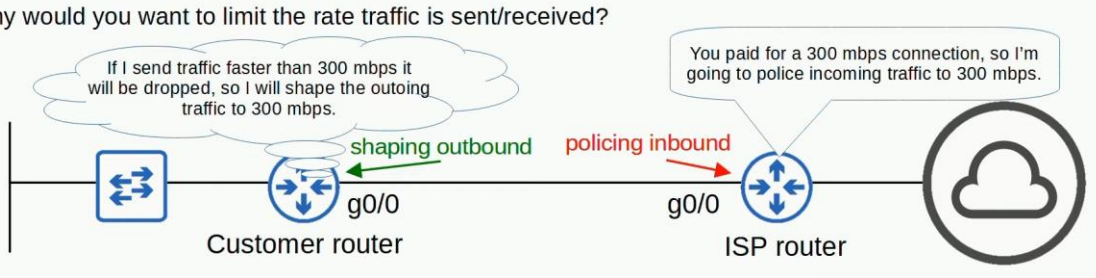


Shaping and Policing

- Traffic **shaping** and **policing** are both used to control the rate of traffic.
- **Shaping** buffers traffic in a queue if the traffic rate goes over the configured rate.
- **Policing** drops traffic if the traffic rate goes over the configured rate.
  → 'Burst' traffic over the configured rate is allowed for a short period of time.
  → This accommodates data applications which typically are 'bursty' in nature. Instead of a constant stream of data, they send data in bursts.
  → The amount of burst traffic allowed is configurable.

*policing also has the option of re-marking the traffic instead of dropping it.

- In both cases, classification can be used to allow for different rates for different kinds of traffic.
- Why would you want to limit the rate traffic is sent/received?



Routing Packets: Default Gateway

- End hosts like PC1 and PC4 can send packets directly to destinations in their connected network.
  → PC1 is connected to 192.168.1.0/24, PC4 is connected to 192.168.4.0/24.
- To send packets to destinations outside of their local network, they must send the packets to their **default gateway**.

PC1 (Linux) Config:

```
iface eth0 inet static
        address 192.168.1.10/24
        gateway 192.168.1.1
```

PC4 (Linux) Config:

```
iface eth0 inet static
        address 192.168.4.10/24
        gateway 192.168.4.4
```

- The **default gateway** configuration is also called a **default route**.
  → It is a route to 0.0.0.0/0 = all netmask bits set to 0. Includes all addresses from 0.0.0.0 to 255.255.255.255.
- End hosts usually have no need for any more specific routes.
  → They just need to know: to send packets outside of my local network, I should send them to my default gateway.

Src. IP: 192.168.1.10
Dst. IP: 192.168.4.10
Dst. MAC: R1 **G0/2** MAC
Src. MAC: PC1 **eth0** MAC

*to learn R1 G0/2's MAC address, PC1 will first send an ARP request to 192.168.1.1.

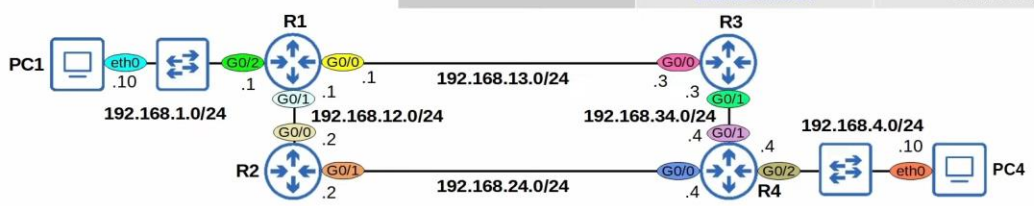
The **default route** is the *least specific* route possible, because it includes all IP addresses.
0.0.0.0/0 = 4,294,967,296 IP addresses
A /32 route (ie. Local route) is the *most specific* route possible, because it specifies only one IP address.
192.168.1.1/32 = 1 IP address

Static Route Configuration

- Each router in the path needs two routes: a route to 192.168.1.0/24 and a route to 192.168.4.0/24.
  → This ensures **two-way reachability** (PC1 can send packets to PC4, PC4 can send packets to PC1).
- R1 already has a **Connected route** to 192.168.1.0/24. R4 already has a **Connected route** to 192.168.4.0/24.
  → The other routes must be manually configured (using **Static routes**).

*routers don't need routes to all networks in the path to the destination.
→ R1 doesn't need a route to 192.168.34.0/24.
→ R4 doesn't need a route to 192.168.13.0/24.

Router	Destination	Next-Hop
R1	192.168.1.0/24	Connected
	192.168.4.0/24	192.168.13.3
R3	192.168.1.0/24	192.168.13.1
	192.168.4.0/24	192.168.34.4
R4	192.168.1.0/24	192.168.34.3
	192.168.4.0/24	Connected

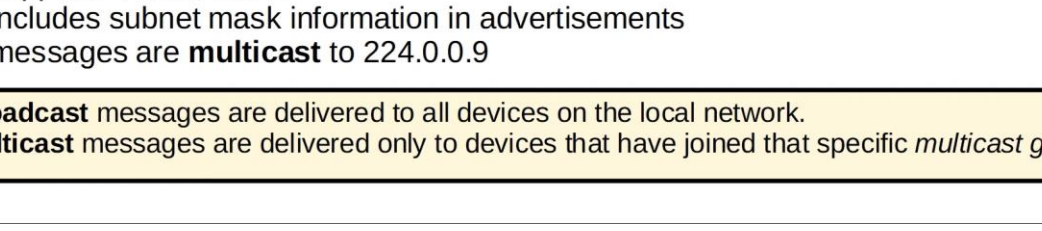


Static Route Configuration with exit-interface

```
R2(config)# ip route 192.168.1.0 255.255.255.0 g0/0
R2(config)# ip route 192.168.4.0 255.255.255.0 g0/1 192.168.24.4

R2(config)# do show ip route
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
!some code output omitted

Gateway of last resort is not set

S     192.168.1.0/24 is directly connected, GigabitEthernet0/0
S     192.168.4.0/24 [1/0] via 192.168.24.4, GigabitEthernet0/1
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/0
L        192.168.12.2/32 is directly connected, GigabitEthernet0/0
      192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.24.0/24 is directly connected, GigabitEthernet0/1
L        192.168.24.2/32 is directly connected, GigabitEthernet0/1
```

```
R2(config)# ip route ip-address netmask exit-interface
R2(config)# ip route ip-address netmask exit-interface next-hop
```

- Static routes in which you specify only the *exit-interface* rely on a feature called **Proxy ARP** to function.
- This is usually not a problem, but generally you can stick to *next-hop* or *exit-interface next-hop*.
- Neither is 'better' than the other: use which you prefer.



RIP

- **Routing Information Protocol** (industry standard)
- Distance vector IGP (uses routing-by-rumor logic to learn/share routes)
- Uses hop count as its metric. One router = one hop (bandwidth is irrelevant!)
- The maximum hop count is **15** (anything more than that is considered unreachable)
- Has three versions:
  RIPv1 and **RIPv2**, used for IPv4
  RIPng (RIP Next Generation), used for IPv6
- Uses two message types:
  Request: To ask RIP-enabled neighbor routers to send their routing table
  Response: To send the local router's routing table to neighboring routers
- By default, RIP-enabled routers will share their routing table every 30 seconds

RIPv1 and RIPv2

- **RIPv1:**
  -only advertises *classful* addresses (Class A, Class B, Class C)
  -doesn't support VLSM, CIDR
  -doesn't include subnet mask information in advertisements (Response messages)
  10.1.1.0/24 will be become 10.0.0.0 (Class A address, so assumed to be /8)
  172.16.192.0/18 will become 172.16.0.0 (Class B address, so assumed to be /16)
  192.168.1.4/30 will become 192.168.1.0 (Class C address, so assumed to be /24)
  -messages are broadcast to 255.255.255.255
- **RIPv2:**
  -supports VLSM, CIDR
  -includes subnet mask information in advertisements
  -messages are **multicast** to 224.0.0.9

Broadcast messages are delivered to all devices on the local network.
Multicast messages are delivered only to devices that have joined that specific *multicast group.*

The network command

- The **network** command tells the router to:

  -look for interfaces with an IP address that is in the specified range

  -active RIP on the interfaces that fall in the range

  -form adjacencies with connected RIP neighbors

  -advertise **the network prefix of the interface** (NOT the prefix in the **network** command)

- The OSPF and EIGRP **network** commands operate in the same way.

EIGRP

- **Enhanced Interior Gateway Routing Protocol**
- Was Cisco proprietary, but Cisco has now published it openly so other vendors can implement it on their equipment.
- Considered an 'advanced' / 'hybrid' distance vector routing protocol.
- Much faster than RIP in reacting to changes in the network.
- Does not have the 15 'hop-count' limit of RIP.
- Sends messages using multicast address 224.0.0.10.
- Is the only IGP that can perform **unequal**-cost load-balancing (by default it performs ECMP load-balancing over 4 paths like RIP)

show ip protocols

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

Routing Protocol is "eigrp 1"
  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(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 172.16.1.14
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.1.0/28
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.0.12.2            90      00:00:23
    10.0.13.2            90      00:00:23
  Distance: internal 90 external 170
```

Router ID order of priority:
1) Manual configuration
2) Highest IP address on a loopback interface
3) Highest IP address on a physical interface

```
R1(config-router)#eigrp router-id ?
  A.B.C.D  EIGRP Router-ID in IP address format

R1(config-router)#eigrp router-id 1.1.1.1
```

Boson Exam Environment — v3.18 :: Custom Exam

File Exam Feedback Help

ExSim-Max for Cisco 200-301 CCNA

Correct

Explanation:
A router uses administrative distance (AD) values to determine route selection when multiple routes to the same destination network are received, and each of these routes is received from a different routing protocol. Lower ADs are preferred over higher ADs. The following list contains the most commonly used ADs:

Route Source	AD
Directly connected route	0
Static route	1
EIGRP summary route	5
eBGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
iBGP	200
Unknown	255

A router uses metrics to determine route selection when multiple routes to the same destination network are received, and all of these routes are received from the same routing protocol. Each routing protocol uses different metrics. For example, Routing Information Protocol (RIP) uses hop count as a metric, Open Shortest Path First (OSPF) uses cost as a metric, and Enhanced Interior Gateway Routing Protocol (EIGRP) uses bandwidth and delay by default as a composite metric. When a routing protocol contains multiple routes to the same destination network, a router prefers the route with the lowest metric.

A router uses prefix lengths to determine route selection when multiple routes to different destination networks are received, regardless of the routing protocol. When multiple routes to overlapping networks exist, a router will prefer the most specific route, which is the route with the longest prefix match. For example, if a router has a packet destined to 10.1.1.1, it will prefer a route to 10.1.1.0/24 over a route to 10.1.1.0/16 and it will prefer a route to 10.1.1.0/30 over a route to 10.1.1.0/24.

Reference:
Cisco: Route Selection in Cisco Routers

Previous Next Show Answer Notes Print Question Review End and Grade

What is routing?

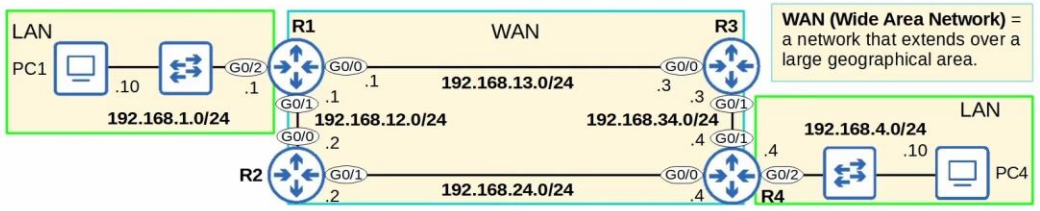
- **Routing** is the process that routers use to determine the path that IP packets should take over a network to reach their destination.
  → Routers store routes to all of their known destinations in a **routing table**.
  → When routers receive packets, they look in the **routing table** to find the best route to forward that packet.

- There are two main routing methods (methods that routers use to learn routes):
  Dynamic Routing: Routers use *dynamic routing protocols* (ie. OSPF) to share routing information with each other automatically and build their routing tables.
  → We will cover this later in the course.

  Static Routing: A network engineer/admin manually configures routes on the router.
  → We will cover this in the next video.

next-hop = the next router in the path to the destination.

- A **route** tells the router: *to send a packet to destination X, you should send the packet to* ***next-hop*** *Y.*
  → or, if the destination is directly connected to the router, *send the packet directly to the destination.*
  → or, if the destination is the router's own IP address, *receive the packet for yourself (don't forward it).*



WAN (Wide Area Network) = a network that extends over a large geographical area.

Routing Table (show ip route)

```
R1# show ip route
```

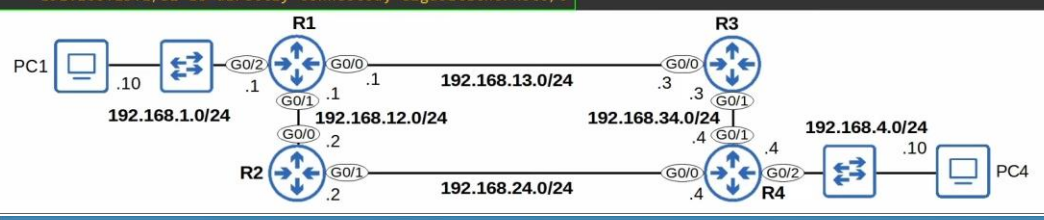
Use the command **show ip route** to view the routing table.

```
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
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/2
L        192.168.1.1/32 is directly connected, GigabitEthernet0/2
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/1
L        192.168.12.1/32 is directly connected, GigabitEthernet0/1
      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.13.0/24 is directly connected, GigabitEthernet0/0
L        192.168.13.1/32 is directly connected, GigabitEthernet0/0
```

The Codes legend in the output of **show ip route** lists the different protocols which routers can use to learn routes.

- **L - local**
  → A route to the actual IP address configured on the interface. (with a /32 netmask)
- **C - connected**
  → A route to the network the interface is connected to. (with the actual netmask configured on the interface)

When you configure an IP address on an interface and enable it with **no shutdown**, 2 routes (per interface) will automatically be added to the routing table:
→ a **connected** route
→ a **local** route

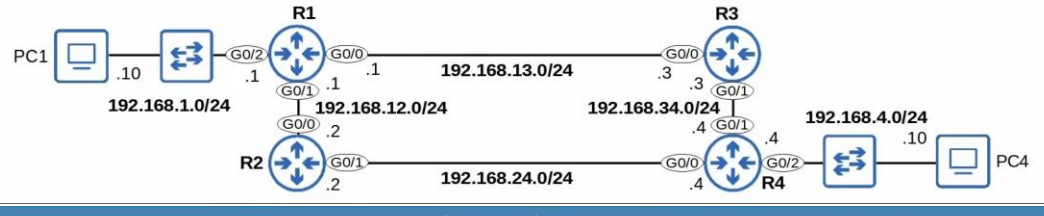


Connected and Local routes

```
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/2
L        192.168.1.1/32 is directly connected, GigabitEthernet0/2
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/1
L        192.168.12.1/32 is directly connected, GigabitEthernet0/1
      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.13.0/24 is directly connected, GigabitEthernet0/0
L        192.168.13.1/32 is directly connected, GigabitEthernet0/0
```

- A **connected** route is a route to the network the interface is connected to.
- R1 G0/2 IP = 192.168.1.1/24
- Network Address = 192.168.1.0/24
- It provides a route to all hosts in that network (ie. 192.168.1.10, 192.168.1.100, 192.168.1.232, etc.)
- R1 knows: "If I need to send a packet to any host in the 192.168.1.0/24 network, I should send it out of G0/2".

- A **local** route is a route to the exact IP address configured on the interface.
- A /32 netmask is used to specify the exact IP address of the interface.
  →/32 means all 32 bits are 'fixed', they can't change.
- Even though R1's G0/2 is configured as 192.168.1.1/24, the connected route is to 192.168.1.1/32.
- R1 knows: "If I receive a packet destined for this IP address, the message is for me".



Connected and Local routes

192 . 168 . 1 . 0 /24

255 . 255 . 255 . 0

=FIXED (can't change)

=not fixed

```
C        192.168.1.0/24 is directly connected, GigabitEthernet0/2
```

- **192.168.1.0/24** *matches* 192.168.1.0 ~ 192.168.1.255.
  → If R1 receives a packet with a destination in that range, it will send the packet out of G0/2.

A route **matches** a packet's destination if the packet's destination IP address is part of the network specified in the route.

192.168.1.2 = **match**
→Send packet out of G0/2
192.168.1.7 = **match**
→Send packet out of G0/2
192.168.1.89 = **match**
→Send packet out of G0/2
192.168.2.1 = **no match**
→Send the packet using a different route, or drop the packet if there is no matching route.

Route Selection

```
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/2
L        192.168.1.1/32 is directly connected, GigabitEthernet0/2
```

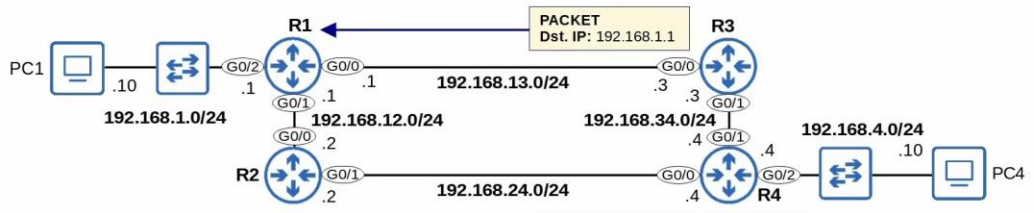
- A packet destined for **192.168.1.1** is matched by both routes:
  192.168.1.0/24
  192.168.1.1/32

When R1 receives a packet destined for 192.168.1.1, it will select the route to 192.168.1.1/32.
→R1 will receive the packet for itself, rather than forward it out of G0/2.
Local route = keep the packet, don't forward

- Which route will R1 use for a packet destined for 192.168.1.1?
  →It will choose the **most specific** matching route.

- The route to **192.168.1.0/24** includes 256 different IP addresses (192.168.1.0 – 192.168.1.255)
- The route to **192.168.1.1/32** includes only 1 IP address (192.168.1.1)
  →This route is more **specific**.

- **Most specific** matching route = the matching route with the **longest prefix length**.



Spanning Tree Versions

Industry standards (IEEE)

Spanning Tree Protocol (802.1D)
- The original STP
- All VLANs share one STP instance.
- Therefore, cannot load balance.

Rapid Spanning Tree Protocol (802.1w)
- Much faster at converging/adapating to network changes than 802.1D
- All VLANs share one STP instance.
- Therefore, cannot load balance.

Multiple Spanning Tree Protocol (802.1s)
- Uses modified RSTP mechanics.
- Can group multiple VLANs into different instances (ie. VLANs 1-5 in instance 1, VLANs 6-10 in instance 2) to perform load balancing.

Cisco versions

Per-VLAN Spanning Tree Plus (PVST+)
- Cisco's upgrade to 802.1D
- Each VLAN has its own STP instance.
- Can load balance by blocking different ports in each VLAN.

Rapid Per-VLAN Spanning Tree Plus (Rapid PVST+)
- Cisco's upgrade to 802.1w
- Each VLAN has its own STP instance.
- Can load balance by blocking different ports in each VLAN.

Rapid Spanning Tree Protocol

Similarities between STP and RSTP:

- RSTP serves the same purpose as STP, blocking specific ports to prevent Layer 2 loops.
- RSTP elects a root bridge with the same rules as STP.
- RSTP elects root ports with the same rules as STP.
- RSTP elects designated ports with the same rules as STP.

Rapid Spanning Tree Port States

STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Discarding	NO/YES	NO	NO	Stable
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable

- If a port is administratively disabled (**`shutdown`** command) = discarding state

Rapid Spanning Tree Port Roles

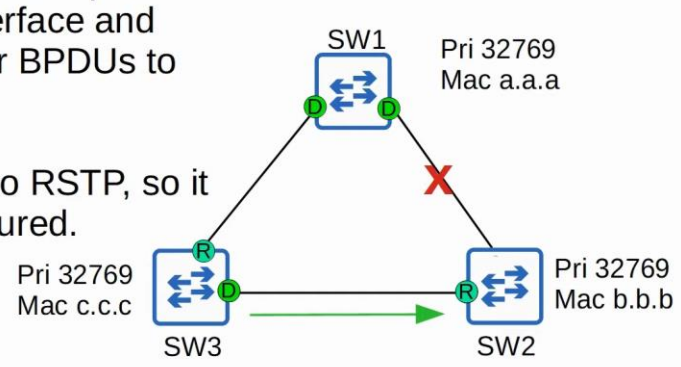
- The **root port** role remains unchanged in RSTP.
  → The port that is closest to the root bridge becomes the root port for the switch.
  → The root bridge is the only switch that doesn't have a root port.
- The **designated port** role remains unchanged in RSTP.
  → The port on a segment (collision domain) that sends the best BPDU is that segment's designated port (only one per segment)
- The **non-designated port** role is split into two separate roles in RSTP:

  the **alternate port** role

  the **backup port** role

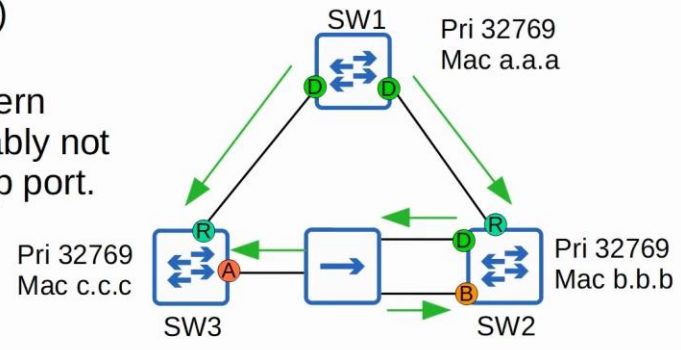
RSTP: BackboneFast functionality

- One more STP optional feature that was built into RSTP is **BackboneFast**.
- BackboneFast allows SW3 to expire the made age timers on its interface and rapidly forward the superior BPDUs to SW2.
- This functionality is built into RSTP, so it does not need to be configured.



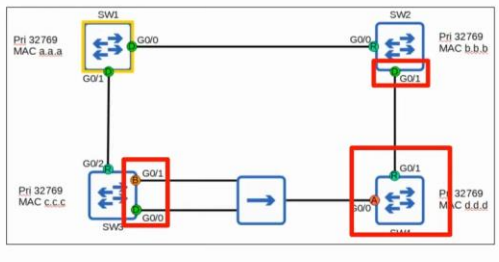
RSTP: Backup port role

- The RSTP **backup** port role is a discarding port that receives a superior BPDU from another interface on the same switch.
- This only happens when two interfaces are connected to the same collision domain (via a hub)
- Hubs are not used in modern networks, so you will probably not encounter an RSTP backup port.
- Function as a backup for a designated port.



Rapid Spanning Tree Protocol

```
SW4#show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    32769
              Address     aaaa.aaaa.aaaa
              Cost        8
              Port        2 (GigabitEthernet0/1)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     dddd.dddd.dddd
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Altn BLK 4         128.1    P2p
Gi0/1               Root FWD 4         128.2    P2p
```



Rapid STP is compatible with Classic STP. The interface(s) on the Rapid STP-enabled switch connected to the Classic STP-enabled switch will operate in Classic STP mode (timers, blocking → listening → learning → forwarding process, etc).

Things we'll cover

- Comparison of STP versions (standard vs Cisco)
- Rapid PVST+
  - → RSTP port states (discarding, learning, forwarding)
  - → RSTP port roles (root, designated, alternate, backup)
  - → STP optional features built into in RSTP (UplinkFast, BackboneFast, PortFast)
  - → RSTP BPDU (sent by all switches, not just the root bridge)
  - → RSTP link types (edge, point-to-point, shared)

Simple Network Management Protocol

- SNMP is an industry-standard framework and protocol that was originally released in 1988.
  RFC 1065 – Structure and identification of management information for TCP/IP-based internets
  RFC 1066 – Management information base for network management of TCP/IP-based internets
  RFC 1067 – A simple network management protocol

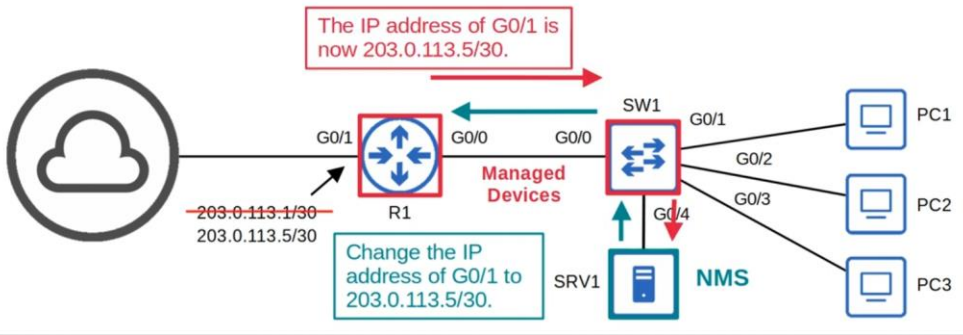
SNMPv1

- Don't let the 'Simple' in the name fool you!
- SNMP can be used to monitor the status of devices, make configuration changes, etc.
- There are two main types of devices in SNMP:
  1) Managed Devices
  → These are the devices being managed using SNMP.
  → For example, network devices like routers and switches.

  2) Network Management Station (NMS)
  → The device/devices managing the managed devices.
  → This is the SNMP 'server'.

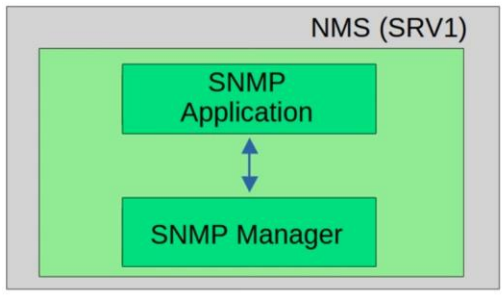
SNMP Operations

- There are three main operations used in SNMP.
  1) Managed devices can notify the NMS of events.
  2) The NMS can ask the managed devices for information about their current status.
  3) The NMS can tell the managed devices to change aspects of their configuration.



SNMP Components

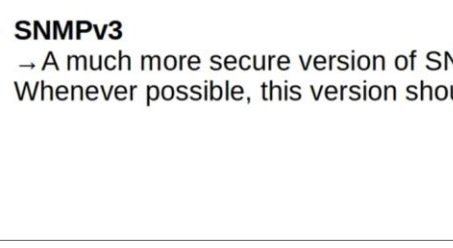
SNMP Components

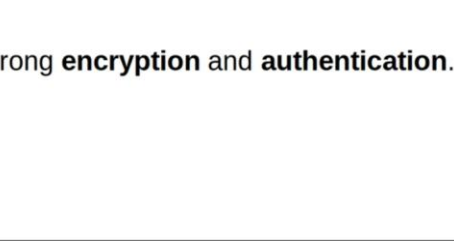


- The **SNMP Manager** is the software on the NMS that interacts with the managed devices.
  → It receives notifications, sends requests for information, sends configuration changes, etc.
- The **SNMP Application** provides an interface for the network admin to interact with.
  → Displays alerts, statistics, charts, etc.

SNMP Components

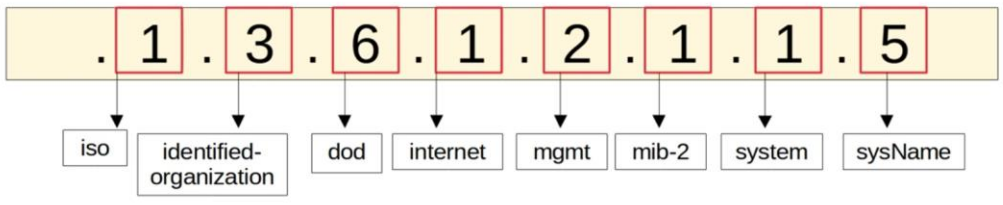
- The **SNMP Agent** is the SNMP software running on the managed devices that interacts with the SNMP Manager on the NMS.
  → It sends notifications to/receives messages from the NMS.
- The **Management Information Base (MIB)** is the structure that contains the variables that are managed by SNMP.
  → Each variable is identified with an Object ID (OID)
  → Example variables: Interface status, traffic throughput, CPU usage, temperature, etc.

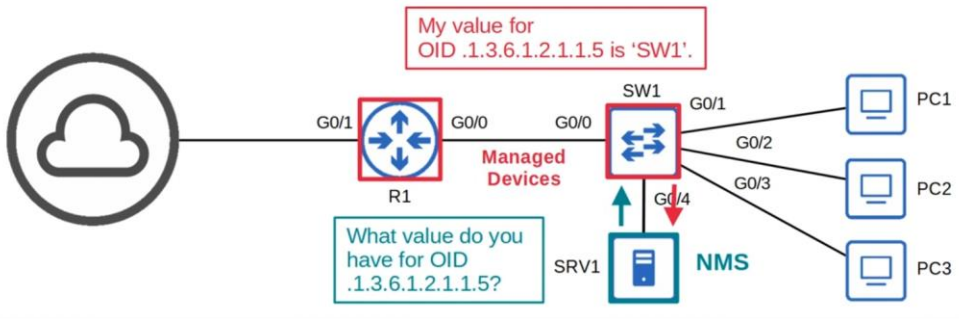




SNMP OIDs

- SNMP Object IDs are organized in a hierarchical structure.





SNMP Versions

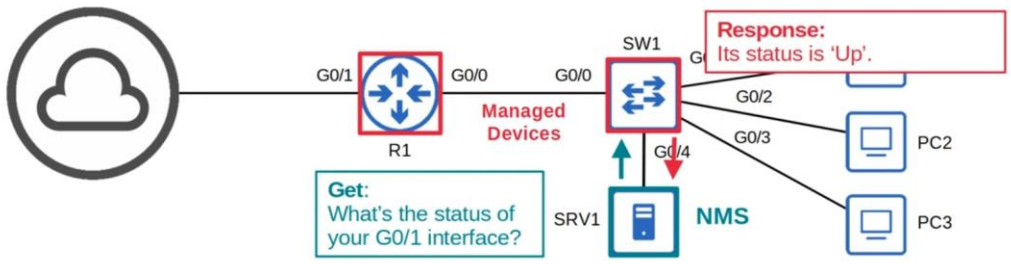
- Many versions of SNMP have been proposed/developed, however only three major versions have achieved wide-spread use:
- **SNMPv1**
  → The original version of SNMP.
- **SNMPv2c**
  → Allows the NMS to retrieve large amounts of information in a single request, so it is more efficient.
  → 'c' refers to the 'community strings' used as passwords in SNMPv1, removed from SNMPv2, and then added back for SNMPv2c.
- **SNMPv3**
  → A much more secure version of SNMP that supports strong **encryption** and **authentication**. Whenever possible, this version should be used!

SNMP Messages

Message Class	Description	Messages
Read	Messages sent by the **NMS** to read information from the **managed devices**. (ie. What's your current CPU usage %?)	*Get* *GetNext* *GetBulk*
Write	Messages sent by the **NMS** to change information on the **managed devices.** (ie. change an IP address)	*Set*
Notification	Messages sent by the **managed devices** to alert the **NMS** of a particular event. (ie. interface going down)	*Trap* *Inform*
Response	Messages sent in response to a previous message/request.	*Response*

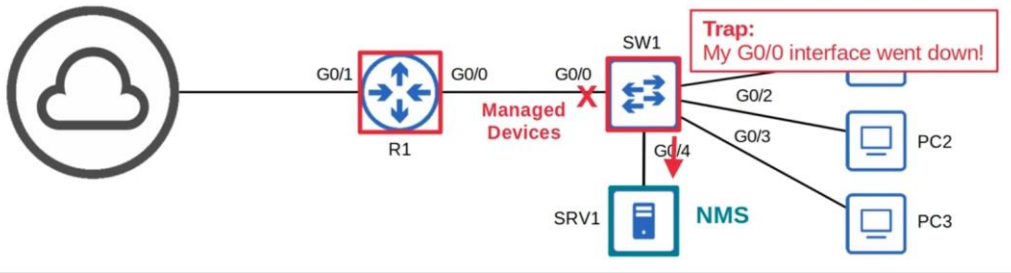
SNMP 'Read' Messages

- **Get**
  →A request sent from the manager to the agent to retrieve the value of a variable (OID), or multiple variables. The agent will send a *Response* message with the current value of each variable.
- **GetNext**
  →A request sent from the manager to the agent to discover the available variables in the MIB.
- **GetBulk**
  →A more efficient version of the **GetNext** message (introduced in SNMPv2).



SNMP 'Notification' Messages

- **Trap**
  →A notification sent from the agent to the manager. The manager does not send a Response message to acknowledge that it received the Trap, so these messages are 'unreliable'.
- **Inform**
  →A notification message that is acknowledged with a Response message.
  →Originally used for communications between managers, but later updates allow agents to send Inform messages to managers, too.



SNMPv2c Configuration

```
R1(config)#snmp-server contact jeremy@jeremysitlab.com
R1(config)#snmp-server location Jeremy's House
```

Optional information

```
R1(config)#snmp-server community Jeremy1 ro

R1(config)#snmp-server community Jeremy2 rw
```

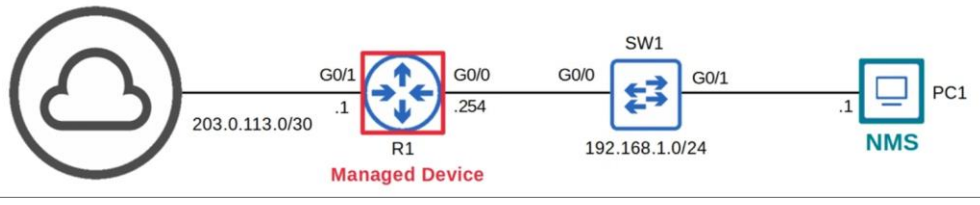
Configure the SNMP *community strings* (passwords)
ro = read only = no Set messages
rw = read/write = can use Set messages

```
R1(config)#snmp-server host 192.168.1.1 version 2c Jeremy1
```

Specify the NMS, version, and community

```
R1(config)#snmp-server enable traps snmp linkdown linkup
R1(config)#snmp-server enable traps config
```

Configure the Trap types to send to the NMS



SNMP Summary

- SNMP helps manage devices over a network.
- **Managed Devices** are the devices being managed using SNMP, such as network devices (routers, switches, firewalls)
- **Network Management Stations (NMS)** are the SNMP 'servers' that manage the devices.
  - → NMS receives notifications from managed devices
  - → NMS changes settings on managed devices
  - → NMS checks status of managed devices
- Variables such as interface status, temperature, traffic load, host name, etc. are stored in the Management Information Base (MIB) and identified using Object IDs (OIDs)
- Main SNMP versions: SNMPv1, SNMPv2c, SNMPv3
- SNMP messages: Get, GetNext, GetBulk, Set, Trap, Inform, Response

Telnet Configuration

```
SW1(config)#enable secret ccna
SW1(config)#username jeremy secret ccna
SW1(config)#access-list 1 permit host 192.168.2.1
SW1(config)#line vty 0 15
SW1(config-line)#login local
SW1(config-line)#exec-timeout 5 0
SW1(config-line)#transport input telnet
SW1(config-line)#access-class 1 in
```

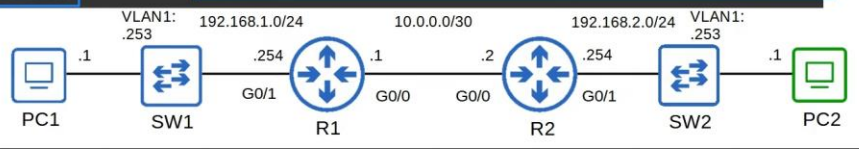
If an enable password/secret isn't configured, you won't be able to access privileged exec mode when connecting via Telnet.

Configure an ACL to limit which devices can connect to the *VTY lines*.

Telnet/SSH access is configured on the VTY lines. There are 16 lines available, so up to 16 users can be connected at once. (VTY stands for Virtual TeleType)

`transport input telnet` allows only Telnet connections.
`transport input ssh` allows only SSH connections.
`transport input telnet ssh` allows both.
`transport input all` allows all connections.
`transport input none` allows no connections.

Apply the ACL to the VTY lines.
***`access-class`** applies an ACL to the VTY lines,
`ip access-group` applies an ACL to an interface.



SSH Configuration: RSA Keys

- To enable and use SSH, you must generate an RSA public and private key pair.
- The keys are used for data encryption/decryption, authentication, etc.

```
SW1(config)#ip domain name jeremysitlab.com
SW1(config)#crypto key generate rsa
The name for the keys will be: SW1.jeremysitlab.com
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

SW1(config)#
*Feb 21 04:22:35.778: %SSH-5-ENABLED: SSH 1.99 has been enabled

SW1(config)#do show ip ssh
SSH Enabled - version 1.99
Authentication methods:publickey,keyboard-interactive,password
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded): SW1.jeremysitlab.com
[output omitted]
```

The **FQDN** of the device is used to name the RSA keys.
FQDN = Fully Qualified Domain Name (host name + domain name)

Generate the RSA keys.
`crypto key generate rsa modulus` *`length`*
is an alternate method.
*length must be 768 bits or greater for SSHv2

SSH Configuration: VTY Lines

```
SW1(config)#enable secret ccna
SW1(config)#username jeremy secret ccna
SW1(config)#access-list 1 permit host 192.168.2.1
SW1(config)#ip ssh version 2
SW1(config)#line vty 0 15
SW1(config-line)#login local
SW1(config-line)#exec-timeout 5 0
SW1(config-line)#transport input ssh
SW1(config-line)#access-class 1 in
```

(optional, but recommended) Restrict SSH to version 2 only.

Configure all VTY lines, just like Telnet.

Enable local user authentication.
*you cannot use **`login`** for SSH, only **`login local`**.

(optional, but recommended) Configure the exec timeout.

Best practice is to limit VTY line connections to SSH only.

(optional, but recommended) Apply the ACL to restrict VTY line connections.

SSH Configuration

1) Configure host name
2) Configure DNS domain name
3) Generate RSA key pair
4) Configure enable PW, username/PW
5) Enable SSHv2 (only)
6) Configure VTY lines

```
Router(config)#crypto key generate rsa

% Please define a hostname other than Router.

Router(config)#hostname R2

R2(config)#crypto key generate rsa

% Please define a domain-name first.

R2(config)#ip domain name jeremysitlab.com

R2(config)#crypto key generate rsa
The name for the keys will be: R2.jeremysitlab.com
[output omitted]
```

Connect: **ssh -l** *username ip-address* OR **ssh** *username@ip-address*

You have to know how to configure SSH for the CCNA exam, so make sure to do the practice lab!

Command Summary

```
SW1# show version
SW1# show ip ssh
SW1(config)# ip default-gateway ip-address
SW1(config)# line con 0
SW1(config)# line vty 0 15
SW1(config)# crypto key generate rsa
SW1(config)# ip ssh version 2
SW1(config-line)# login [local]
SW1(config-line)# transport input [protocols | all | none]
SW1(config-line)# exec-timeout minutes seconds
SW1(config-line)# access-class acl in
```

```
> telnet ip-address
> ssh -l username ip-address
> ssh username@ip-address
```

Network Redundancy

- Redundancy is an essential part of network design.
- Modern networks are expected to run 24/7/365. Even a short downtime can be disastrous for a business.
- If one network component fails, you must ensure that other components will take over with little or no downtime.
- As much as possible, you must implement redundancy at every possible point in the network.

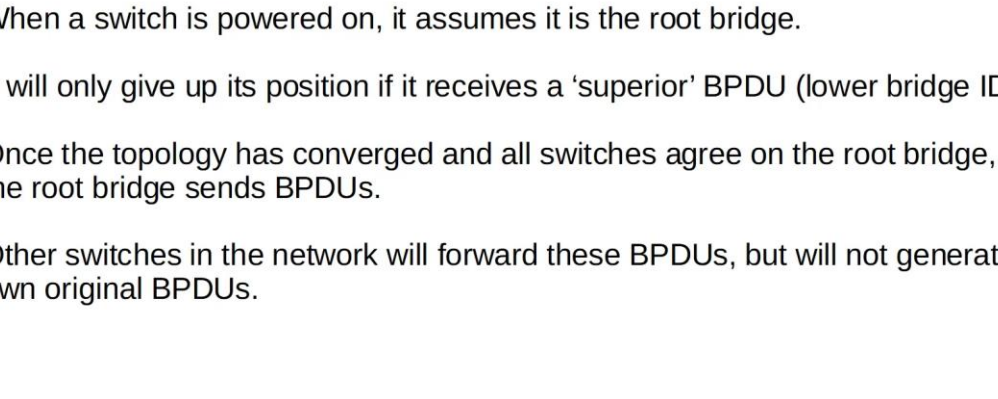
Spanning Tree Protocol

- 'Classic Spanning Tree Protocol' is IEEE 802.1D.
- Switches from ALL vendors run STP by default.
- STP prevents Layer 2 loops by placing redundant ports in a blocking state, essentially disabling the interface.
- These interfaces act as backups that can enter a forwarding state if an active (=currently forwarding) interface fails.
- Interfaces in a forwarding state behave normally. They send and receive all normal traffic.
- Interfaces in a blocking state only send or receive STP messages (called BPDUs = Bridge Protocol Data Units).

Spanning Tree Protocol

- Switches use one field in the STP BPDU, the **Bridge ID** field, to elect a **root bridge** for the network.
- The switch with the lowest **Bridge ID** becomes the **root bridge**.
- ALL ports on the **root bridge** are put in a forwarding state, and other switches in the topology must have a path to reach the root bridge.

Spanning Tree Protocol



Cisco switches use a version of STP called **PVST** (Per-VLAN Spanning Tree). PVST runs a separate STP 'instance' in each VLAN, so in each VLAN different interfaces can be forwarding/blocking.

Spanning Tree Protocol

Bridge Priority				Extended System ID (VLAN ID)											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1

= 28673 (16384 + 8192 + 4096 + 1)

The STP bridge priority can only be changed in units of 4096.

The valid values you can configure are:
0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, or 61440.
The Extended System ID will then be added to this number to make the total bridge priority.

Spanning Tree Protocol

- When a switch is powered on, it assumes it is the root bridge.
- It will only give up its position if it receives a 'superior' BPDU (lower bridge ID).
- Once the topology has converged and all switches agree on the root bridge, only the root bridge sends BPDUs.
- Other switches in the network will forward these BPDUs, but will not generate their own original BPDUs.

Spanning Tree Protocol

Speed	STP Cost
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

Spanning Tree Protocol

1) One switch is elected as the root bridge. All ports on the root bridge are **designated ports** (forwarding state). Root bridge selection:
   1: Lowest bridge ID

2) Each remaining switch will select ONE of its interfaces to be its **root port** (forwarding state)**.** Ports across from the root port are always **designated** ports.
   Root port selection:
   1: Lowest root cost
   2: Lowest neighbor bridge ID
   3: Lowest neighbor port ID

3) Each remaining collision domain will select ONE interface to be a **designated port** (forwarding state). The other port in the collision domain will be **non-designated** (blocking)
   Designated port selection:
   1: Interface on switch with lowest root cost
   2: Interface on switch with lowest bridge ID

Spanning Tree Port States

STP Port State	Stable/Transitional
Blocking	Stable
Listening	Transitional
Learning	Transitional
Forwarding	Stable

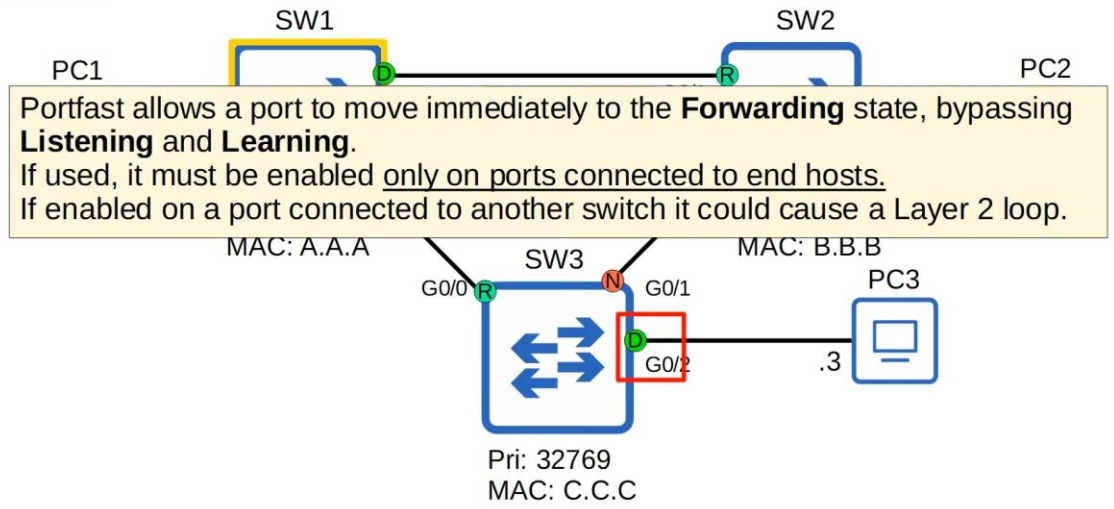
- Root/Designated ports remain stable in a **Forwarding** state.
- Non-designated ports remain stable in a **Blocking** state.
- **Listening** and **Learning** are transitional states which are passed through when an interface is activated, or when a **Blocking** port must transition to a Forwarding state due to a change in the network topology.

Spanning Tree Timers

STP Timer	Purpose	Duration
Hello	How often the root bridge sends hello BPDUs	2sec
Forward delay	How long the switch will stay in the Listening and Learning states (each state is 15 seconds = total 30 seconds)	15sec
Max Age	How long an interface will wait after ceasing to receive Hello BDPUs to change the STP topology.	20sec (10* hello)

Spanning Tree Optional Features (STP Toolkit)

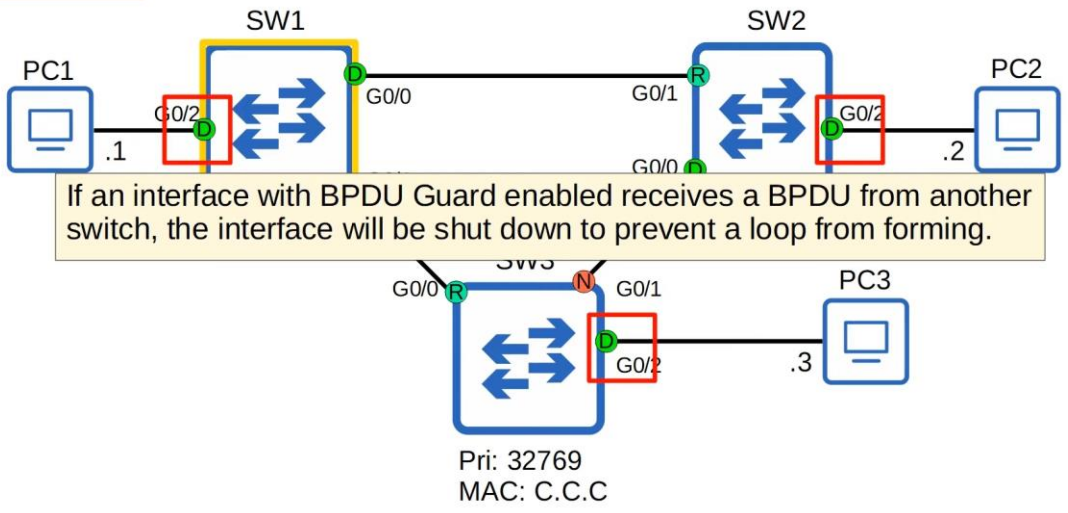
Portfast



Portfast allows a port to move immediately to the **Forwarding** state, bypassing **Listening** and **Learning**.
If used, it must be enabled only on ports connected to end hosts.
If enabled on a port connected to another switch it could cause a Layer 2 loop.

Spanning Tree Optional Features (STP Toolkit)

BPDU Guard



If an interface with BPDU Guard enabled receives a BPDU from another switch, the interface will be shut down to prevent a loop from forming.

Spanning Tree Optional Features (STP Toolkit)

BPDU Guard

```
SW1(config)#interface g0/2
SW1(config-if)#spanning-tree bpduguard enable
SW1(config-if)#
```

You can also enable BPDU Guard with the following command:

`SW1(config)# spanning-tree portfast bpduguard default`

This enables BPDU Guard on all Portfast-enabled interfaces.

Spanning Tree Optional Features (STP Toolkit)

BPDU Guard

2.5 Describe the need for and basic operations of Rapid PVST+ Spanning Tree Protocol and identify

You probably don't have to know these STP optional features (or others such as UplinkFast, Backbone Fast, etc) for the CCNA. But make sure you know **Portfast** and **BPDU Guard**. If you want to read more about the others just in case, do a Google search.

Root Guard	If you enable **root guard** on an interface, even if it receives a superior BPDU (lower bridge ID) on that interface, the switch will not accept the new switch as the root bridge. The interface will be disabled.
Loop Guard	If you enable **loop guard** on an interface, even if the interface stops receiving BPDUs, it will not start forwarding. The interface will be disabled.

IPv4 Address Classes

- The IANA (Internet Assigned Numbers Authority) assigns IPv4 addresses/networks to companies based on their size.
- For example, a very large company might receive a **class A** or **class B** network, while a small company might receive a **class C** network.
- However, this led to many wasted IP addresses.

CIDR (Classless Inter-Domain Routing)

- When the Internet was first created, the creators did not predict that the Internet would become as large as it is today.
- This resulted in wasted address space like the examples I showed you (there are many more examples).
- The IETF (Internet Engineering Task Force) introduced CIDR in 1993 to replace the 'classful' addressing system.

Subnets/Hosts (Class B)

Prefix Length	Number of Subnets	Number of Hosts
/17	2	32766
/18	4	16382
/19	8	8190
/20	16	4094
/21	32	2044
/22	64	1022
/23	128	510
/24	256	254

Prefix Length	Number of Subnets	Number of Hosts
/25	512	126
/26	1024	62
/27	2048	30
/28	4096	14
/29	8192	6
/30	16384	2
/31	32768	0 (2)
/32	65536	0 (1)

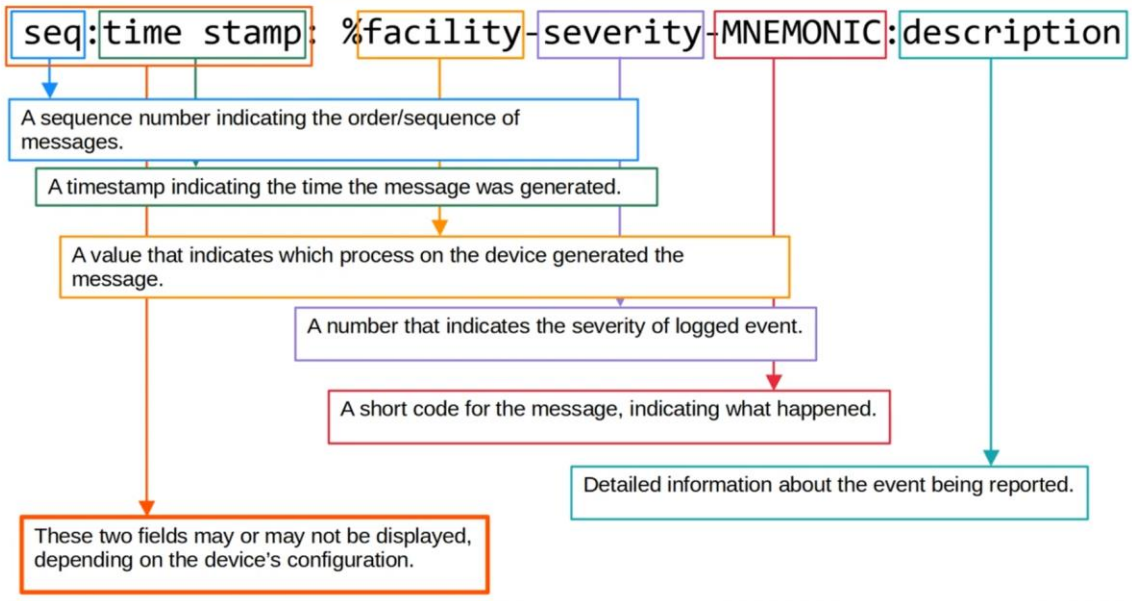
Syslog

- Syslog is an industry standard protocol for message logging.
- On network devices, Syslog can be used to log events such as changes in interface status (up⇨ down), changes in OSPF neighbor status (up⇨ down), system restarts, etc.
- The messages can be displayed in the CLI, saved in the device's RAM, or sent to an external Syslog server.

```
R1(config)#int g0/0
R1(config-if)#no shutdown
R1(config-if)#
*Feb 11 03:02:55.304: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Feb 11 03:02:56.305: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
```

- Logs are essential when troubleshooting issues, examining the cause of incidents, etc.
- Syslog and SNMP are both used for monitoring and troubleshooting of devices. They are complementary, but their functionalities are different.

Syslog Message Format



Syslog Severity Levels

Level	Keyword	Description
0	**Emergency**	System is unusable
1	**Alert**	Action must be taken immediately
2	**Critical**	Critical conditions
3	**Error**	Error conditions
4	**Warning**	Warning conditions
5	**Notice**	Normal but significant condition (**Notification**)
6	**Informational**	Informational messages
7	**Debugging**	Debug-level messages

Every **A**wesome **C**isco **E**ngineer **W**ill **N**eed **I**ce cream **D**aily

Syslog Configuration

```
!configure logging to the console line
R1(config)#logging console 6

!configure logging to the vty lines
R1(config)#logging monitor informational

!configure logging to the buffer
R1(config)#logging buffered 8192 6

!configure logging to an external server
R1(config)#logging 192.168.1.100

R1(config)#logging host 192.168.1.100

R1(config)#logging trap debugging
```

logging console *level*
*you can use the level **number** (6) or **keyword** (informational)
*this will enable logging for the *informational* severity and higher

logging monitor *level*
*same points as above about the level

logging buffered [*size*] *level*
*same points as above about the level
*buffer size is in bytes

logging *server-ip*
logging host *server-ip*
*these commands are the same!

logging trap *level*
*same points as above about the level
*this sets the logging level for the external server

logging synchronous

- By default, logging messages displayed in the CLI while you are in the middle of typing a command will result in something like this:

```
R1(config)#exit
R1#show ip in
*Feb 11 09:38:41.607: %SYS-5-CONFIG_I: Configured from console by jeremy on
consoleterface brief
```

- To prevent this, you should use the **logging synchronous** on the appropriate *line*.
(I will talk more about 'line' configuration in the Telnet/SSH video!)

```
R1(config)#line console 0
R1(config-line)#logging synchronous
```

- This will cause a new line to be printed if your typing is interrupted by a message.

```
R1(config)#exit
R1#show ip int
*Feb 11 09:41:00.554: %SYS-5-CONFIG_I: Configured from console by jeremy on console
R1#show ip int
```

show ip int was reprinted on a new line. This makes it easier to continue typing the command.

Syslog Command Summary

```
R1(config)# logging console severity
R1(config)# logging monitor severity
R1(config)# logging buffered [size] severity
R1(config)# logging server-ip
R1(config)# logging host server-ip
R1(config)# logging trap severity
R1# terminal monitor
R1(config-line)# logging synchronous
R1(config)# service timestamps log [datetime | uptime]
R1(config)# service sequence-numbers
```

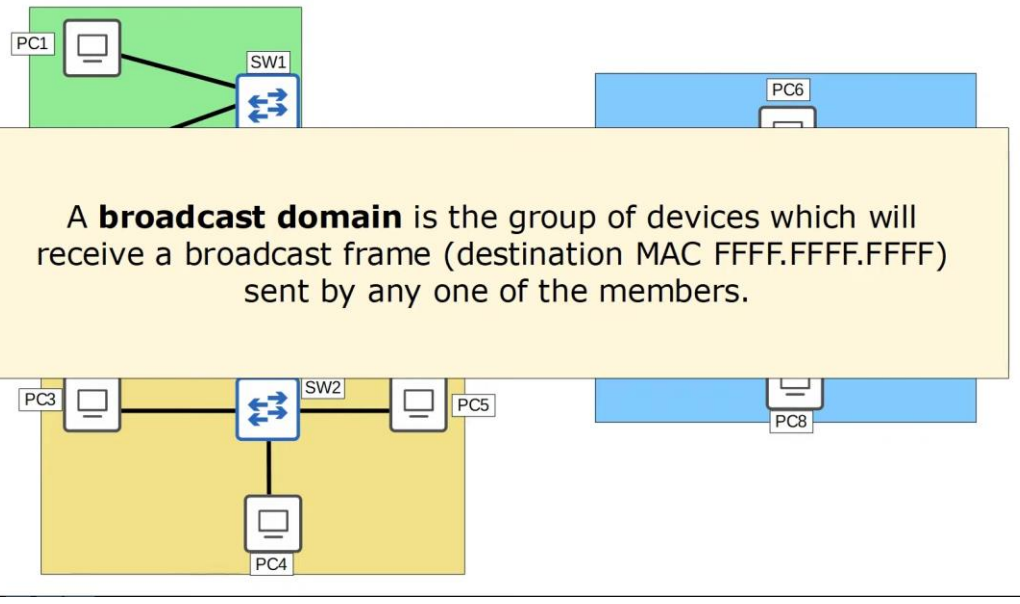
Syslog vs SNMP

- Syslog and SNMP are both used for monitoring and troubleshooting of devices. They are complementary, but their functionalities are different.
- **Syslog** is used for message logging.
  → Events that occur within the system are categorized based on facility/severity and logged.
  → Used for system management, analysis, and troubleshooting.
  → Messages are sent from the devices to the server. The server **can't** actively pull information from the devices (like SNMP **Get**) or modify variables (like SNMP **Set**).
- **SNMP** is used to retrieve and organize information about the SNMP managed devices.
  → IP addresses, current interface status, temperature, CPU usage, etc.
  → SNMP servers can use **Get** to query the clients and **Set** to modify variables on the clients.

What is a LAN?

- Previously I said that a LAN is a group of devices (PCs, servers, routers, switches, etc.) in a single location (home, office, etc.)
- A more specific definition: A LAN is a single **broadcast domain**, including all devices in that broadcast domain.
- A **broadcast domain** is the group of devices which will receive a broadcast frame (destination MAC FFFF.FFFF.FFFF) sent by any one of the members.

LANs/Broadcast Domains



What is a VLAN?

192.168.1.0/24

Performance: Lots of unnecessary broadcast traffic can reduce network performance.

Security: Even within the same office, you want to limit who has access to what.
You can apply security policies on a router/firewall.

Because this is one LAN, PCs can reach each other directly,
without traffic passing through the router.

So, even if you configure security policies,
they won't have any effect.

HR

VLAN Configuration

```
SW1(config)#interface range g1/0 - 3
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
SW1(config-if-range)#interface range g2/0 - 2
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 20
% Access VLAN does not exist. Creating vlan 20
SW1(config-if-range)#interface range g3/0 - 3
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 30
% Access VLAN does not exist. Creating vlan 30
SW1(config-if-range)#
```

An access port is a switchport which belongs to a single VLAN,
and usually connects to end hosts like PCs.

Switchports which carry multiple VLANs are called 'trunk ports'.
(More information on trunks in the next video!)

Trunk Ports

Engineering

R1

Switches will 'tag' all frames that they send over a trunk link.
This allows the receiving switch to know which VLAN the frame belongs to.

VLAN30

Trunk ports = 'tagged' ports

Access ports = 'untagged' ports

SW1

SW2

Sales
192.168.1.128/26

HR
192.168.1.64/26

VLAN20

Engineering
192.168.1.0/26

VLAN Tagging

- There are two main trunking protocols: ISL (Inter-Switch Link) and IEEE 802.1Q. dot1q
- ISL is an old Cisco proprietary protocol created before the industry standard IEEE 802.1Q.
- IEEE 802.1Q is an industry standard protocol created by the IEEE (Institute of Electrical and Electronics Engineers).
- You will probably NEVER use ISL in the real world. Even modern Cisco equipment doesn't support it. For the CCNA, you only need to learn 802.1Q.

802.1Q Tag

Preamble	SFD	Destination	Source	802.1Q	Type

- The 802.1Q tag is inserted between the **Source** and **Type/Length** fields of the Ethernet frame.
- The tag is 4 bytes (32 bits) in length.
- The tag consists of two main fields:
  - Tag Protocol Identifier (TPID)
  - Tag Control Information (TCI)
- The TCI consists of three sub-fields.

802.1Q Tag

802.1Q tag format

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

Native VLAN

- 802.1Q has a feature called the **native VLAN**. (ISL does not have this feature)
- The native VLAN is VLAN 1 by default on all trunk ports, however this can be manually configured on each trunk port.
- The switch does not add an 802.1Q tag to frames in the native VLAN.
- When a switch receives an untagged frame on a trunk port, it assumes the frame belongs to the native VLAN.

Router on a Stick (ROAS)

The subinterface number **does not** have to match the VLAN number. However it is **highly recommended** that they do match, to make it easier to understand.

```
R1(config)#i
R1(config-if
R1(config-if
*Apr 15 04:29:49.681: %L                    face GigabitEthernet0/0, changed state to up
*Apr 15 04:29:50.682: %LINE                 DOWN: Line protocol on Interface GigabitEthernet0/
0, changed state to up
R1(config-if)#interface g0/0.10
R1(config-subif)#encapsulation dot1q 10
R1(config-subif)#ip address 192.168.1.62 255.255.255.192
R1(config-subif)#interface g0/0.20
R1(config-subif)#encapsulation dot1q 20
R1(config-subif)#ip address 192.168.1.126 255.255.255.192
R1(config-subif)#interface g0/0.30
R1(config-subif)#encapsulation dot1q 30
R1(config-subif)#ip address 192.168.1.190 255.255.255.192
R1(config-subif)#
```

Native VLAN on a router (ROAS)

- There are **2 methods** of configuring the native VLAN on a router:

  -Use the command **`encapsulation dot1q`** *`vlan-id`* **`native`** on the router subinterface.

  -Configure the IP address for the native VLAN on the router's physical interface (the **`encapsulation dot1q`** *`vlan-id`* command is not necessary)

Native VLAN on a router (ROAS)

- There are **2 methods** of configuring the native VLAN on a router:

  -Use the command **`encapsulation dot1q`** *`vlan-id`* **`native`** on the router subinterface.

```
R1(config)#int g0/0.10
R1(config-subif)#encapsulation dot1q 10 native
R1(config-subif)#
```

Layer 3 (Multilayer) Switches

- A multilayer switch is capable of both *switching* AND *routing*.
- It is 'Layer 3 aware'.
- You can assign IP addresses to its interfaces, like a router.
- You can create virtual interfaces for each VLAN, and assign IP addresses to those interfaces.
- You can configure routes on it, just like a router.
- It can be used for inter-VLAN routing.