CCNA Notes

Jeremy’s IT Lab
CCNA 200-301
Complete Course 2024

Credit:

YouTube Series | Jeremy's IT Lab - CCNA 200-301
https://www.youtube.com/playlist?list=PLxbwE86jKRgMpuZuLBivzlM8s2Dk5lXBQ

Peter Saumur's Github Notes | Jeremy's IT Lab - CCNA 200-301
https://github.com/psaumur/CCNA_Course_Notes

Table of Contents

1. NETWORKING DEVICES

What is a network?
A computer network is a digital telecommunications network allows NODES to share RESOURCES.
A CLIENT is a device that accesses a service made available by a SERVER.
A SERVER is a device that provides functions or services for CLIENTS.

- Note : The same device can be a CLIENT in some situations and a SERVER in other situations. Ex: A Peer-to-Peer network.

SWITCHES (Level 2):

- provide connectivity to hosts within the same LAN (Local Area Network)
- Have many network interfaces/ports for End Hosts to connect to.
- DO NOT provide connectivity between LANs/over the Internet.

ROUTERS (Level 3):

- have fewer network interfaces than switches.
- are used to provide connectivity BETWEEN LANs.
- are used to send data over the Internet.

FIREWALL (Can be Level 3,4, and 7):

- Firewalls are specialty hardware network security devices that control network traffic entering/exiting your network.
- Can be places "inside" or "outside" the network.
- Monitor and control network traffic based on configured rules.
- Are known as "Next-Generation Firewalls" when they include more modern and advanced filtering capabilities.
- Host-based firewalls are software applications that filter traffic entering and exiting a host machine, like a PC.

2. INTERFACES AND CABLES

SWITCHES provide many PORTS for connectivity (usually 24)
These PORTS tend to be RJ-45 (Registered Jack) ports.

WHAT IS ETHERNET?

- Ethernet is a collection of network protocols/standards.

Why do we need network protocols and standards?

- provide common communication standards over networks.
- provide common hardware standards to allow connectivity between devices.

Connections between devices operates at a set speed.
These speeds are measured in "bits per second" (bps)
A bit is a value of "0" or "1". A byte is 8 bits (0s and 1s)

Size	# of Bits
1 kilobit (Kb)	1,000
1 megabit (Mb)	1,000,000
1 gigabit (Gb)	1,000,000,000
1 terabit (Tb)	1,000,000,000,000

Ethernet standards are:

- Defined in the IEEE 802.3 standard in 1983
- IEEE = Institute of Electrical and Electronics Engineers

ETHERNET STANDARDS (COPPER)

Speed	Common Name	Standard	Cable Type	Max Transmission Distance
10 Mbps	Ethernet	802.3i	10BASE-T	100m Max
100 Mbps	Fast Ethernet	802.3u	100BASE-T	100m Max
1 Gbps	Gigabit Ethernet	802.3ab	1000BASE-T	100m Max
10 Gbps	10 Gigabit Ethernet	802.3an	10GBASE-T	100m Max

BASE = refers to Baseband Signaling
T = Twisted Pair
Most Ethernet uses copper cables.
UTP or Unshielded Twisted Pair (no metallic shield) Twist protects against EMI (Electromagnetic Interference)
Most use 8 wires (4 pairs) however ...
10/100BASE-T = 2 pairs (4 wires)

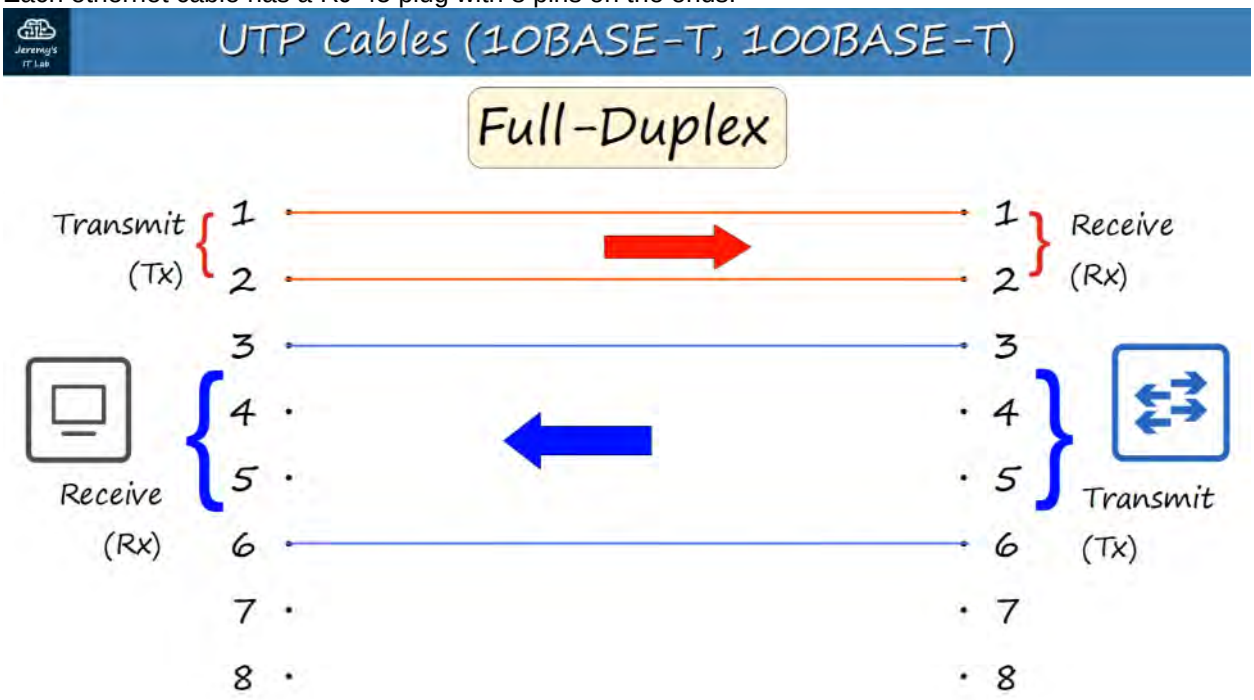
Explanation

Cat (Category) 5e is a kind of copper cable used for Ethernet connections. It supports speeds of up to 1 Gbps and should be a maximum of 100 meters in length to avoid signal attenuation.

Below are a few cable standards:

- **Cat 3**
  - 10 Mbps (10BASE-T)
- **Cat 5**
  - 100 Mbps (100BASE-T)
- **Cat 5e**
  - 1 Gbps (1000BASE-T)
- **Cat6a**
  - 10 Gbps (10GBASE-T)

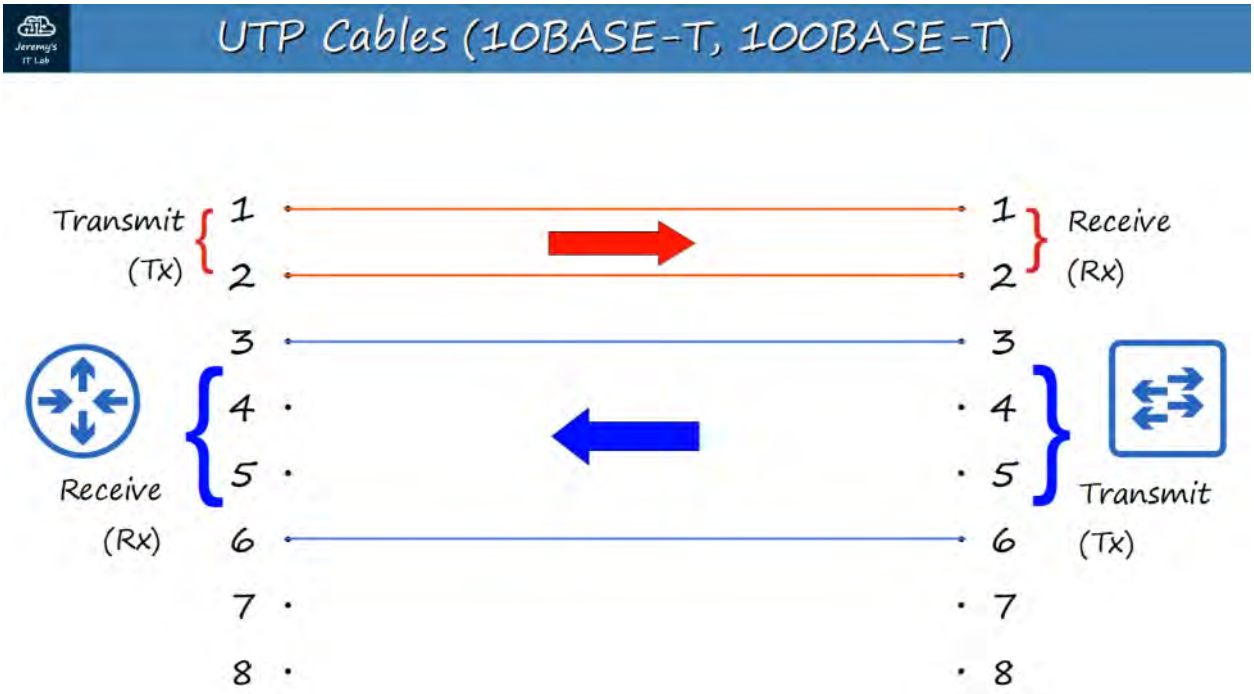
How do devices communicate via their connections?
Each ethernet cable has a RJ-45 plug with 8 pins on the ends.



- PCs Transmit(TX) data on Pins #1-2
- Switches Receive(RX) data on Pins #1-2
- PCs Receive(RC) data on Pins #3,6
- Switches Transmit(TX) data on Pins #3,6

This allows Full-Duplex transmission of data.

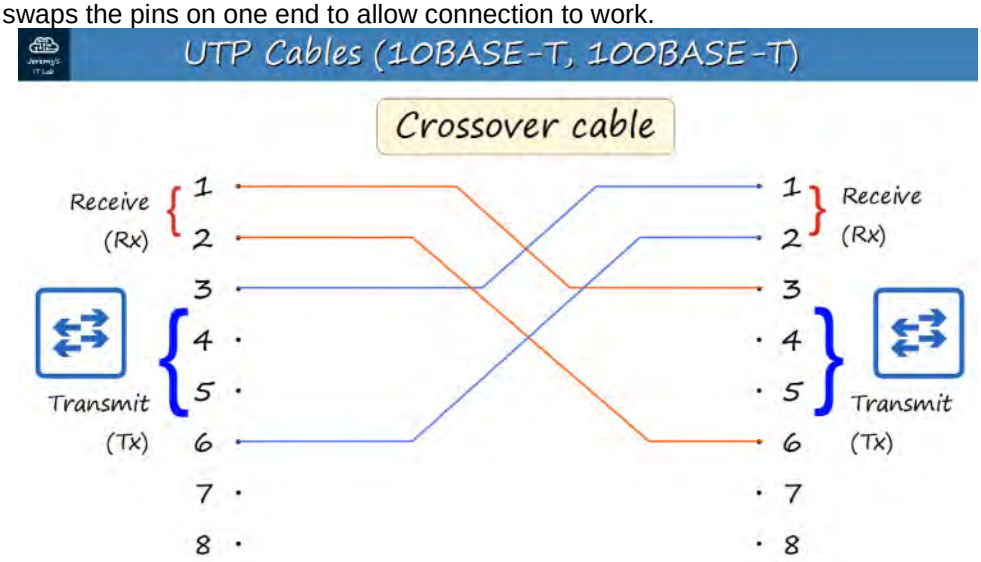
What if a Router / Switch connect?



- Routers Transmit(TX) data on Pins #1-2
- Routers Receive(RX) data on Pins #3,6
- Switches Transmit(TX) data on Pins #3,6
- Switches Receive(RX) data on Pins #1-2

Routers and PCs connect the same way with Switches.
The cable used to connect is called a "Straight-Through" cable.

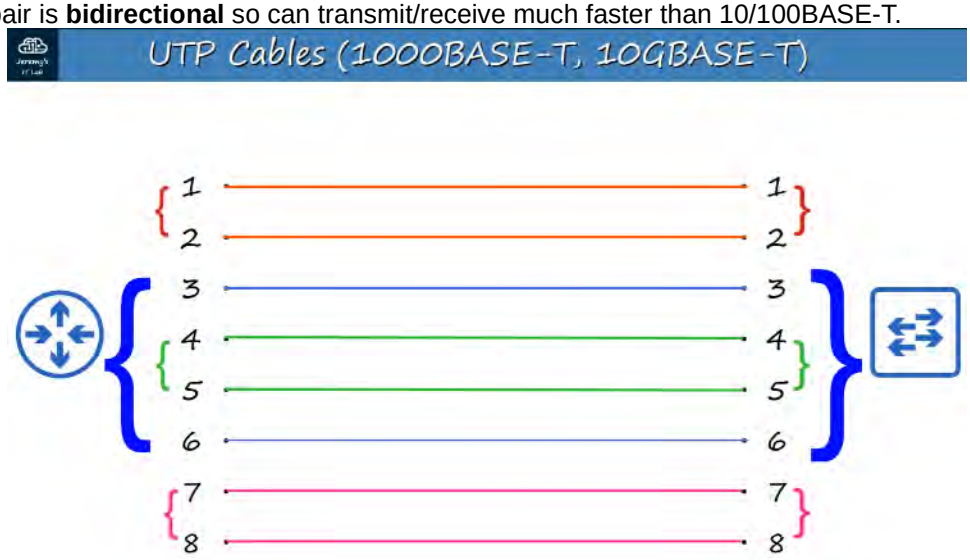
What if we want to connect similar devices to each other?
We CANNOT use a "Straight-Through" cable. We MUST use a "Crossover" cable.
This cable swaps the pins on one end to allow connection to work.



PIN#1 -----> PIN#3 PIN#2 -----> PIN#6
PIN#3 -----> PIN#1 PIN#6 -----> PIN#2

DEVICE TYPE	TRANSMIT (TX) PINS	RECEIVE (RX) PINS
ROUTER	1 and 2	3 and 6
FIREWALL	1 and 2	3 and 6
PC	1 and 2	3 and 6
SWITCH	3 and 6	1 and 2

Most modern equipment now has AUTO MDI-X which **automatically detects** which pins their neighbour is transmitting on and adjust the pins they receive data on.
1000BASE-T/10GBASE-T = 4 pairs (8 wires)
Each wire pair is **bidirectional** so can transmit/receive much faster than 10/100BASE-T.



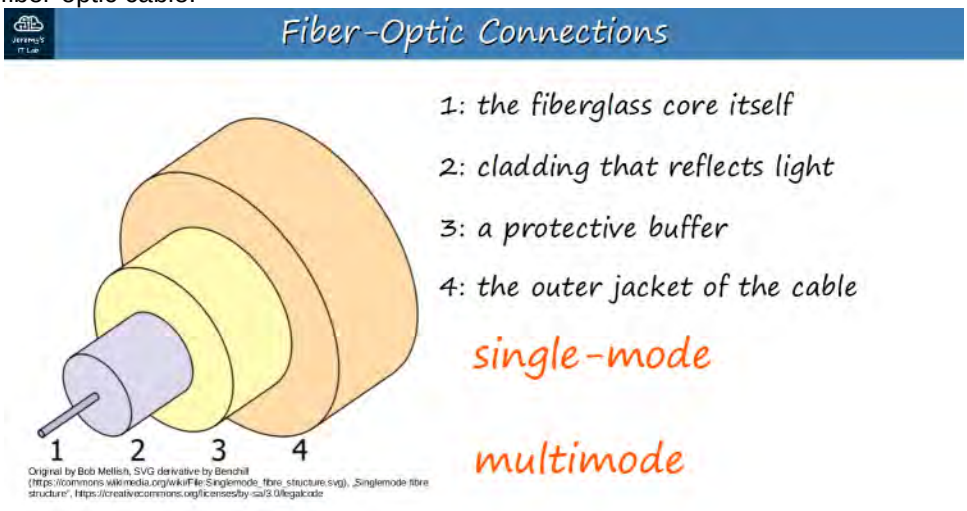
Fiber-Optic Connections:

- Defined in the IEEE 802.3ae standard

SFP Transceiver (Small Form-Factor Pluggable) allows fiber-optic cables to connect to switches/routers.

- Have separate cables to transmit / receive.

4 parts to a fiber-optic cable.

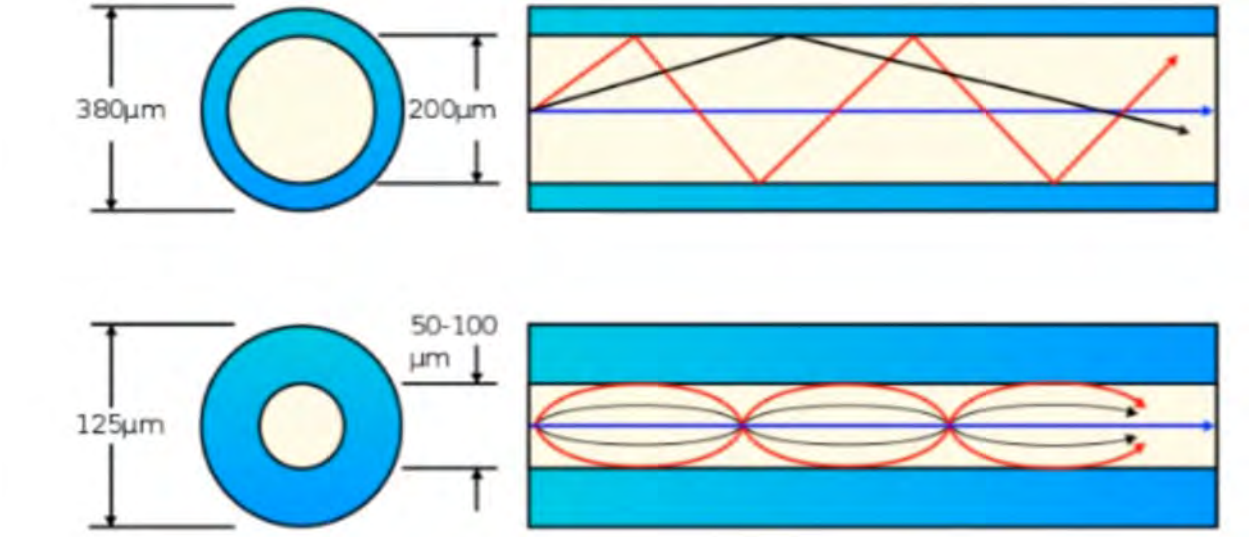


There are TWO types of fiberoptic cable.

Single-Mode:

- Narrower than multimode
- Lighter enters at a single angle (mode) from a laser-based transmitter.
- Allows longer cables than both UTP and multimode fiber.
- More expensive than multimode fiber (due to more expensive laser-based SFP transmitters)

Multimode:



- Core is wider than Single-mode
- Allows multiple angles (modes) of light waves to enter core
- Allows longer cables than UTP but shorter than single-mode
- Cheaper than single-mode fiber (due to cheaper LED-based SFP transmitter)

Fiber Optic Standards:

Speed	Standard	Connection Speed	Mode Support	Max Transmission Distance
1000BASE-LX	802.3z	1 Gbps	Multimode / Single	550 meters (Multi) / 5km (Single)
10GBASE-SR	802.3ae	10 Gbps	Multimode	400 meters
10GBASE-LR	802.3ae	10 Gbps	Single	10 kilometers
10GBASE-ER	802.3ae	10 Gbps	Single	30 kilometers

UTP vs Fiber-Optic Cabling:

UTP are:

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

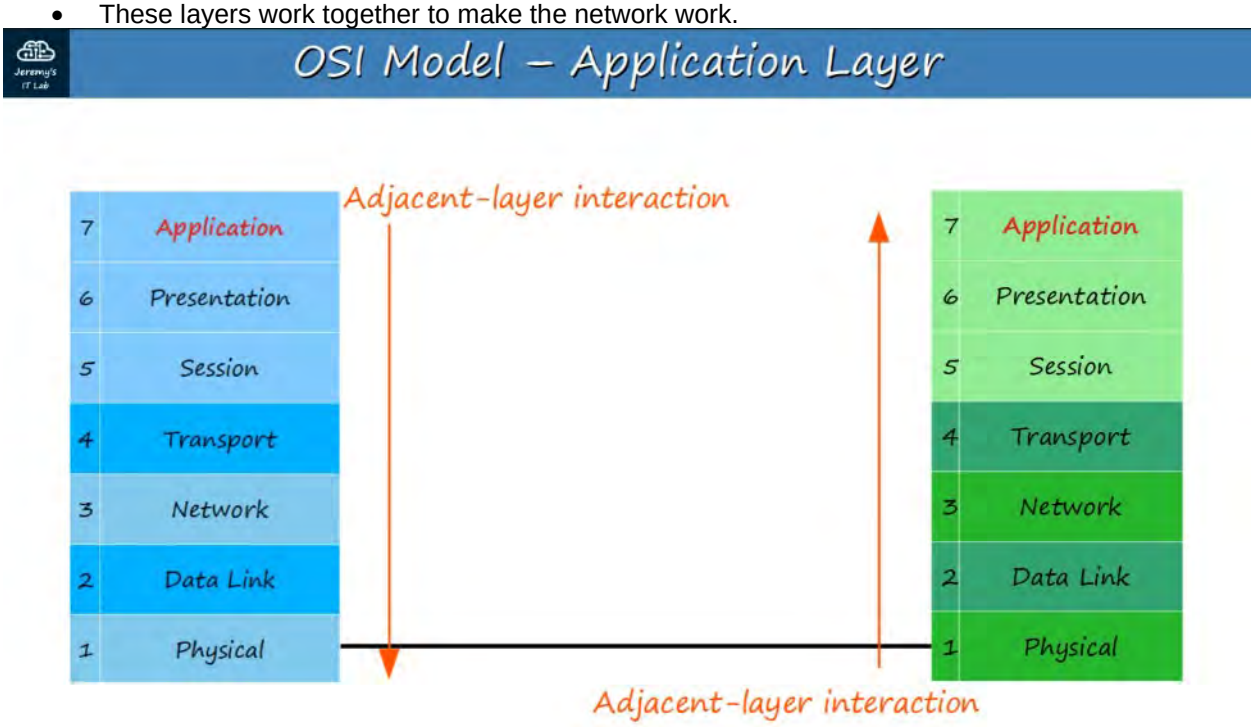
Fiber-Optic:

- 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).

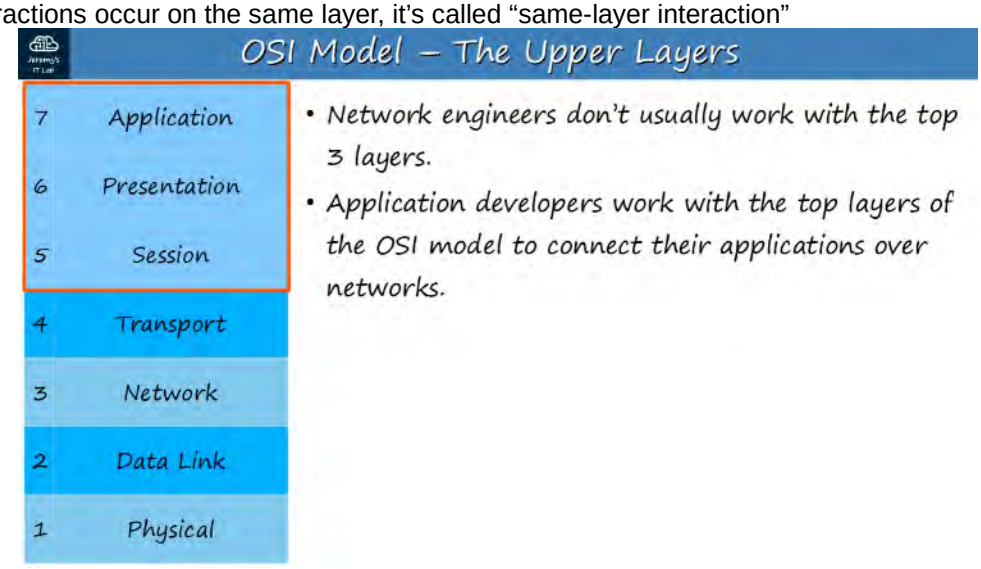
3. OSI MODEL & TCP/IP SUITE

What is a networking model?
Networking models categorize and provide a structure for networking protocols and standards. (Protocols are a set of logical rules defining how network devices and software should work)
OSI MODEL

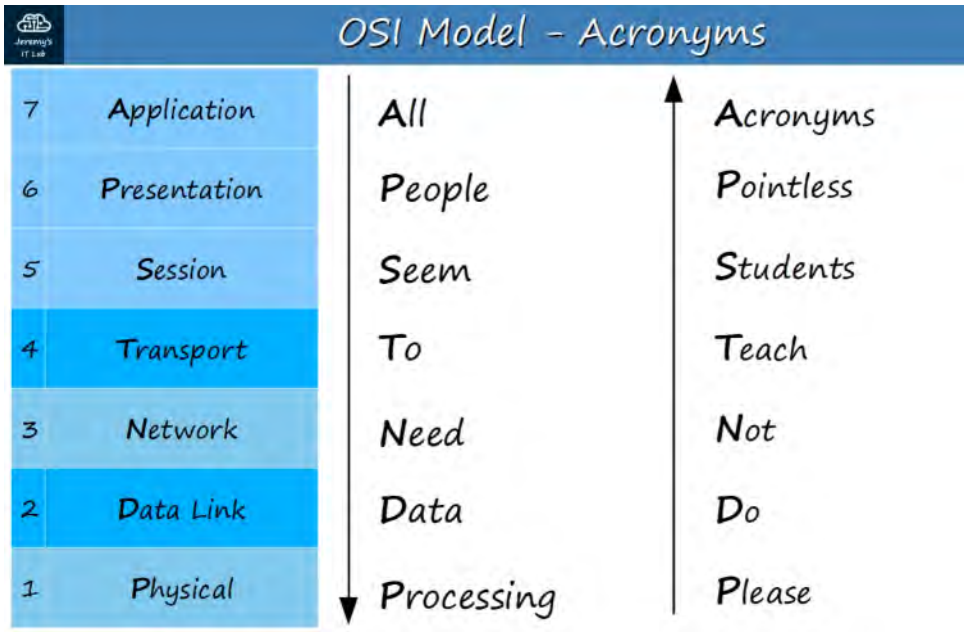
- Open Systems Interconnection Model
- Conceptual model that categorizes and standardizes the different functions in a network.
- Created by the "International Organization for Standardization" (ISO)
- Functions are divided into 7 "Layers"
- These layers work together to make the network work.



As data moves from the top layer, downward, the process is called “encapsulation”
As data moves from the bottom layer, upward, the process is called “de-encapsulation”
When interactions occur on the same layer, it’s called “same-layer interaction”



Mnemonic to help remember the Data Layer Names / Order



The layers are :

7 - APPLICATION

- This Layer is closest to end user.
- Interacts with software applications.
- HTTP and HTTPS are Layer 7 protocols

Functions of Layer 7 include:

- Identifying communication partners
- Synchronizing communication

---

6 - PRESENTATION

- Translates data to the appropriate format (between Application and Network formats) to be sent over the network.

---

5 - SESSION

- Controls dialogues (sessions) between communicating hosts.
- Establishes, manages, and terminates connections between local application and the remote application.

Network engineers don't usually work with the top 3 layers. Application developers work with the top layers of the OSI model to connect their applications over networks.

---

4 - TRANSPORT

- Segments and reassembles data for communication between end hosts.
- Breaks large pieces of data into smaller segments which can be more easily sent over the network and are less likely to cause transmission problems if errors occur.
- Provides HOST-TO-HOST (end to end) communication

When Data from Layer 7-5 arrives, it receives a Layer 4 Header in the Transport layer.
<< DATA + L4 Header >>
This is called a SEGMENT.

---

3 - NETWORK

- Provides connectivity between end hosts on different networks (ie: outside of the LAN).
- Provides logical addressing (IP Addresses).
- Provides path selection between source and destination

- **ROUTERS** operate at Layer 3.

When Data and the Layer 4 Header arrive in the Network Layer, it receives a Layer 3 Header.
<< DATA + L4 Header + L3 Header >>
This is called a **PACKET**.

---

2 - DATA LINK

- Provides NODE-TO-NODE connectivity and data transfer (for example, PC to Switch, Switch to Router, Router to Router)
- Defines how data is formatted for transmission over physical medium (for example, copper UTP cables)
- Detects and (possibly) corrects Physical (Layer 1) errors.
- Uses Layer 2 addressing, separate from Layer 3 addressing.
- **SWITCHES** operate at Layer 2

When the Layer 3 Packet arrives, a Layer 2 Trailer and Header are added.
<< L2 Trailer + DATA + L4 Header + L3 Header + L2 Header >>
This is called a FRAME.
All the steps leading up to transmission is called ENCAPSULATION. When the frame is sent to the receiver, it then goes through the reverse process, DE-ENCAPSULATION, stripping off layers while travelling from OSI Layer 1 to Layer 7.

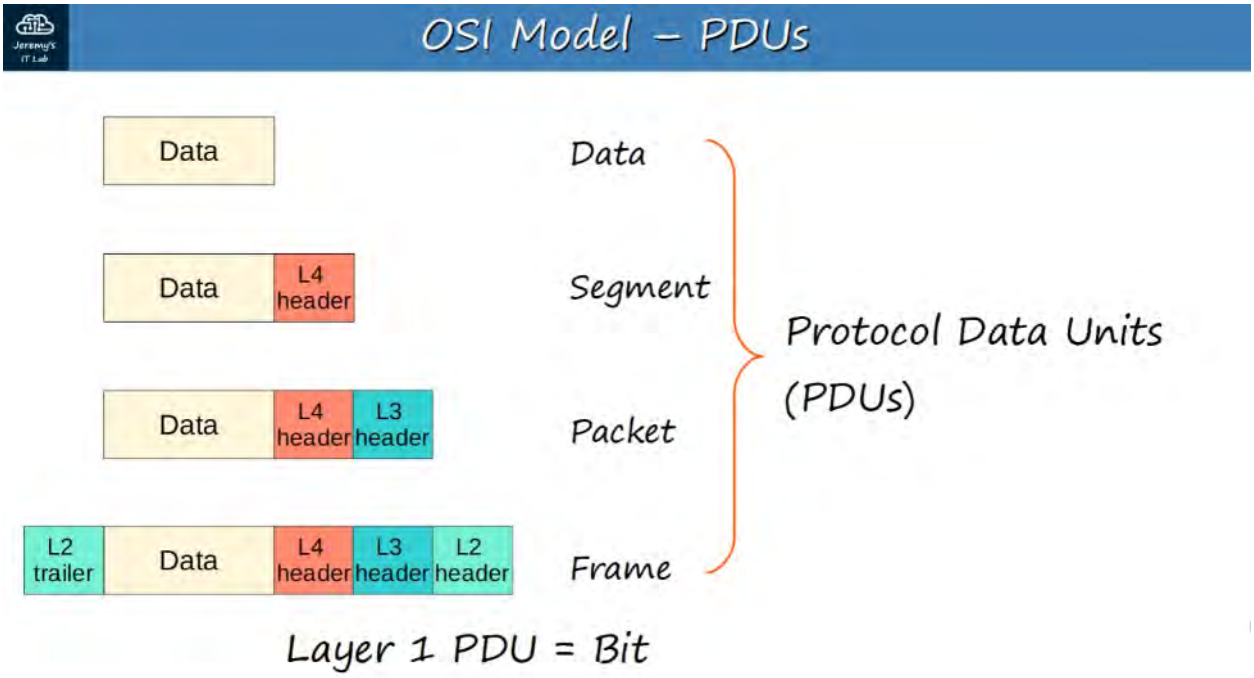
---

1 - PHYSICAL

- Defines physical characteristics of the medium used to transfer data between devices. For example : voltage levels, maximum transmission distances, physical connectors, cable specs.
- Digital bits are converted into electrical (for wired connections) or radio (for wireless connections) signals.
- All of the information in SECTION 2 (NETWORKING DEVICES) is related to the Physical Layer

---

OSI MODEL - PDU's



A PDU is a Protocol Data Unit. Each step of the process is a PDU.

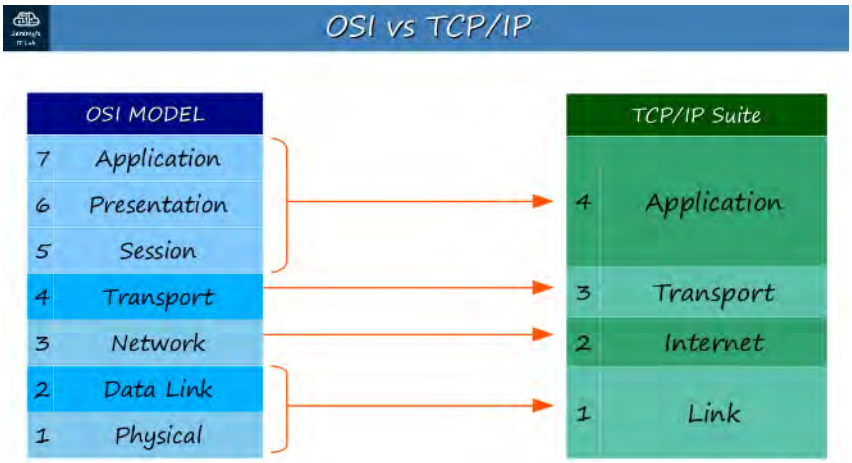
OSI LAYER #	PDU NAME	PROTOCOL DATA ADDED
7-5	DATA	Data

OSI LAYER #	PDU NAME	PROTOCOL DATA ADDED
4	SEGMENT	Layer 4 Header Added
3	PACKET	Layer 3 Header Added
2	FRAME	Layer 2 Trailer and Header Added
1	BIT	0s and 1s Transmission

<< L2 Trailer + DATA + L4 Header + L3 Header + L2 Header >>

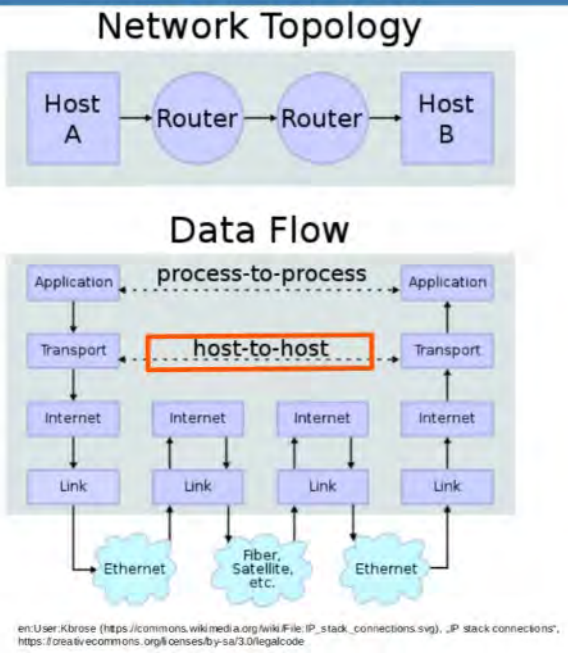
TCP/IP Suite

- 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 US Dept. of Defense through DARPA (Defense Advanced Research Projects Agency).
- Similar structure to the OSI Model, but 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.



Layer Interactions

TCP/IP Suite



en:User:Kbrose (https://commons.wikimedia.org/wiki/File:IP_stack_connections.svg), „IP stack connections", https://creativecommons.org/licenses/by-sa/3.0/legalcode

Adjacent-Layer Interactions:

- Interactions between different layers of the OSI Model on same host.

Example:
Layers 5-7 sending Data to Layer 4, which then adds a Layer 4 header (creating a SEGMENT).
Same-Layer Interactions:

- Interactions between the same Layer on different hosts.
- The concept of Same-Layer interaction allows you to ignore the other layers involved and focus on the interactions between a single layer on different devices.

Example:
The Application Layer of YouTube's web server and your PC's browser.

4. INTRO TO THE CLI

What is a CLI?

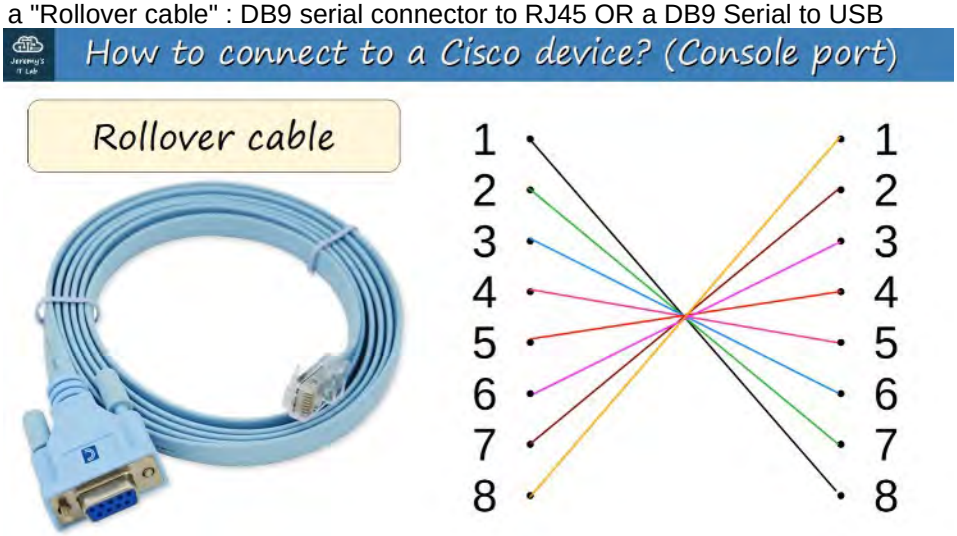
- A "Command-line Interface"
- The interface you use to configure Cisco devices

A GUI is a "Graphical User Interface"

How do you connect to a Cisco Device?

- Console Port : When you first configure a device, you have to connect via the Console Port.

You can use a "Rollover cable" : DB9 serial connector to RJ45 OR a DB9 Serial to USB



How do you actually access the CLI?

- You need to use a TERMINAL EMULATOR (Example: PuTTy is a popular choice) and connect via "Serial" (default settings)

Cisco Default Settings are:

Speed (baud) : 9600 bits/second Data bits: 8 data bits Stop bits: 1 stop bit (sent after 8 data bits are sent)
Parity: None Flow Control: None

When you first enter the CLI you will DEFAULT be in what is called 'User EXEC' mode.
USER EXEC MODE:
(Hostname) > // Prompt looks like THIS //

- User EXEC mode is very limited.
- User can look at some things but can't make ANY changes to the configuration.
- AKA 'User Mode'

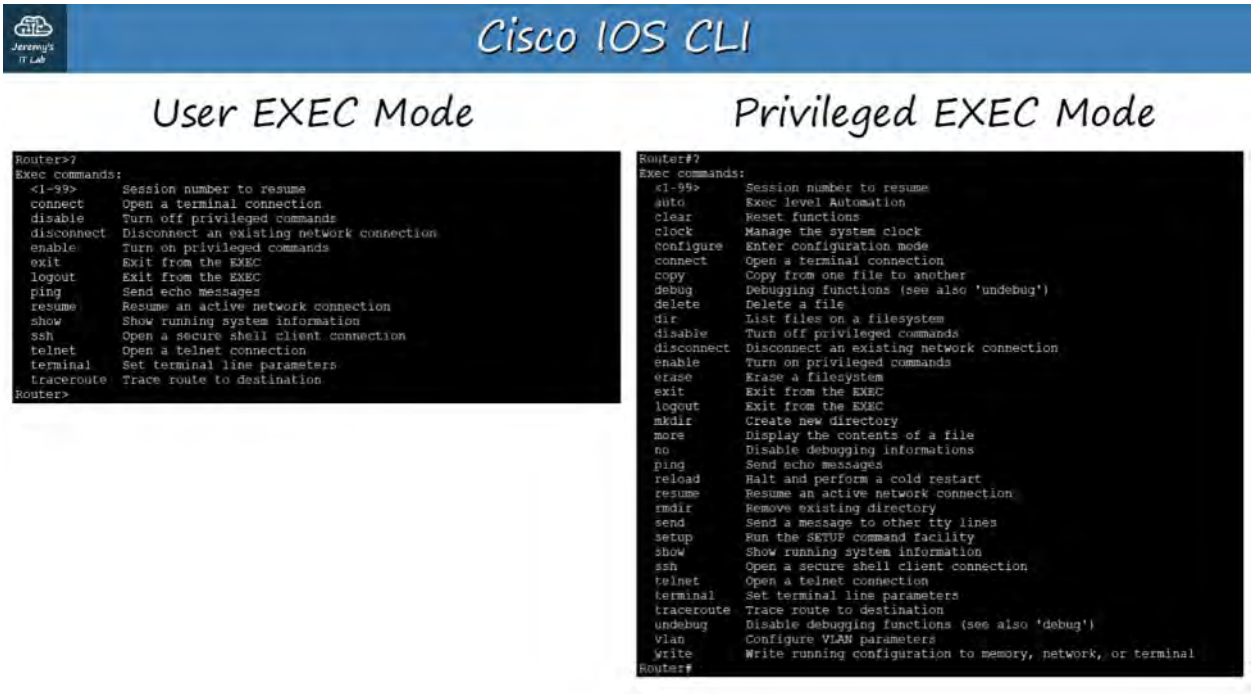
Using the 'enable' command, in User EXEC mode, switches you to 'Privileged EXEC' mode.

PRIVILEGED EXEC MODE:

- Provides complete access to view the device's configuration, restart the device, etc.
- Cannot change the configuration, but can change the time on the device, save the configuration file, etc.

(Hostname)# // Prompt looks like THIS //

USE a Question Mark (?) to view the available commands in ANY mode. Combining ? with a letter or partial command will list all the commands with those letters.



USE the TAB key to complete partially entered commands IF the command exists.

GLOBAL CONFIGURATION MODE:
To enter Global Configuration Mode, enter the command, within Privileged EXEC mode
'configure terminal' (or 'conf t')
Router# configure terminal Router(config) #
Router(config) # run
Router(config) # no
Type 'exit' to drop back into 'Privileged EXEC' mode.

To Enable Password for User EXEC mode:
Router(config)# enable password (password)

- Passwords ARE case-sensitive.

// This command encrypts plain-text passwords, visible in the config files, using simple encryption.
Router(config)# 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.
- The 'enable secret' WILL NOT be effected.

// This command enables passwords for the Privileged EXEC mode.
Router(config)# enable secret (password)
// enable secret will ALWAYS be encrypted (at level 5)

There are TWO separate configuration files kept on the device at once.
Running-config :

- The current, ACTIVE configuration file on the device. As you enter commands in the CLI, you edit the active configuration.

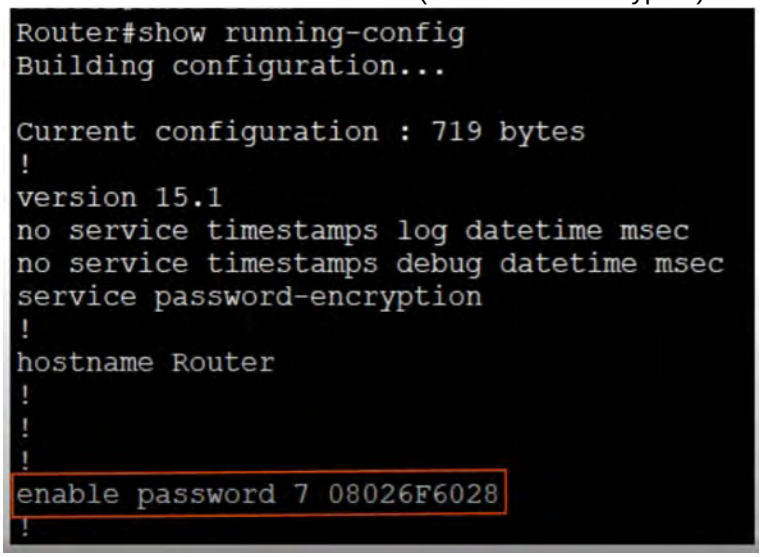
Startup-config :

- The configuration file that will be loaded upon RESTART of the device.

To see the configuration files, inside 'Privileged EXEC' mode:
Router# show running-config // for running config //
OR
Router# show startup-config // for startup config //

To SAVE the Running configuration file, you can:
Router# write Building configuration... [OK]
Router# write memory Building configuration... [OK]
Router# copy running-config startup-config
Destination filename [startup-config]?
Building configuration... [OK]

To encrypt passwords:
Router# conf t
Router(config)# service password-encryption
This makes all current passwords *encrypted*
Future passwords will ALSO be *encrypted*
“Enable secret” will not be effected (it’s ALWAYS encrypted)



Now you will see that the password is no longer in plaintext.
“7” refers to the type of encryption used to encrypt the password. In this case, “7” uses Cisco’s proprietary encryption.
“7” is fairly easy to crack since the encryption is weak.
For BETTER / STRONGER encryption, use “enable secret”

```
Router(config)#enable secret Cisco
Router(config)#do sh run
Building configuration...

Current configuration : 766 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname Router
!
!
!
enable secret 5 $1$mERr$YlCkLMcTYWwkF1Ccndtll.
enable password 7 08026F6028
```

"5" refers to MD5 encryption.
Can still be cracked but it's much much stronger.
Once you use "enable secret" command, this will override "enable password"

To CANCEL or delete a command you entered, use the "no" keyword

```
Router(config)#no service password-encryption
Router(config)#do sh run
Building configuration...

Current configuration : 769 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
enable secret 5 $1$mERr$YlCkLMcTYWwkF1Ccndtll.
enable password 7 08026F6028
!
```

In this instance, disabling "service password-encryption":

- current passwords will NOT be decrypted (unchanged)
- future passwords will NOT be encrypted
- the "enable secret" will not be effected

Modes Review

```
Router>              = user EXEC mode

Router#              = privileged EXEC mode

Router(config)#      = global configuration mode
```

Command Review

```
Router>enable
##used to enter privileged EXEC mode

Router#configure terminal
##used to enter global configuration mode

Router(config)#enable password password
##configures a password to protect privileged exec mode
```

Command Review

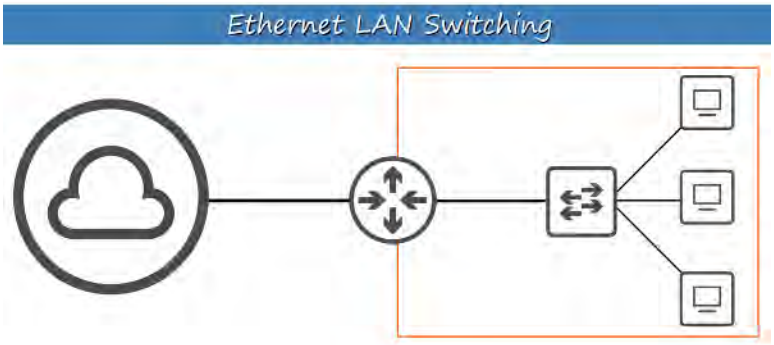
```
Router(config)#service password-encryption
##encrypts the enable password (and other passwords)

Router(config)#enable secret password
##configures a more secure, always-encrypted enable password

Router(config)#run privileged-exec-level-command
##executes a privileged-exec level command from global
configuration mode
```

Command Review

```
Router(config)#no command
##removes the command

Router(config)#show running-config
##displays the current, active configuration file

Router(config)#show startup-config
##displays the saved configuration file which will be loaded if the
device is restarted
```


Command Review
Router(config)#write
##saves the configuration
Router(config)#write memory
##saves the configuration
Router(config)#copy running-config startup-config
##saves the configuration

5. ETHERNET LAN SWITCHING : PART 1

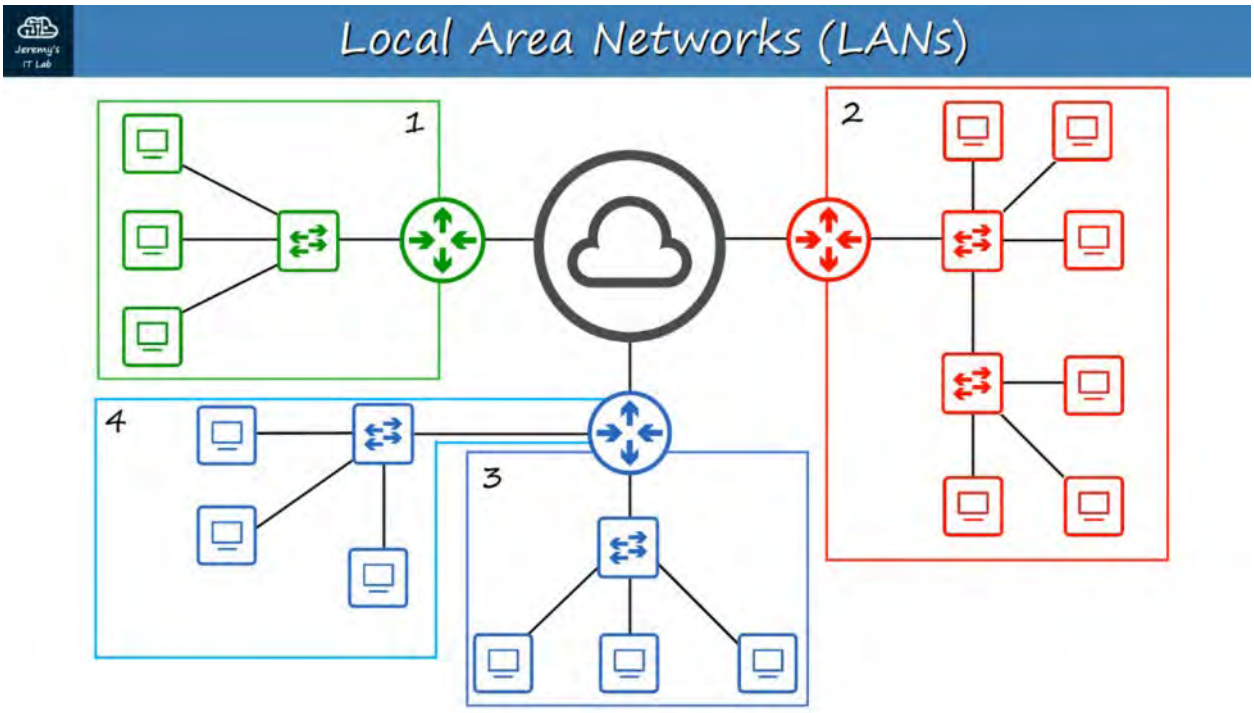


7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

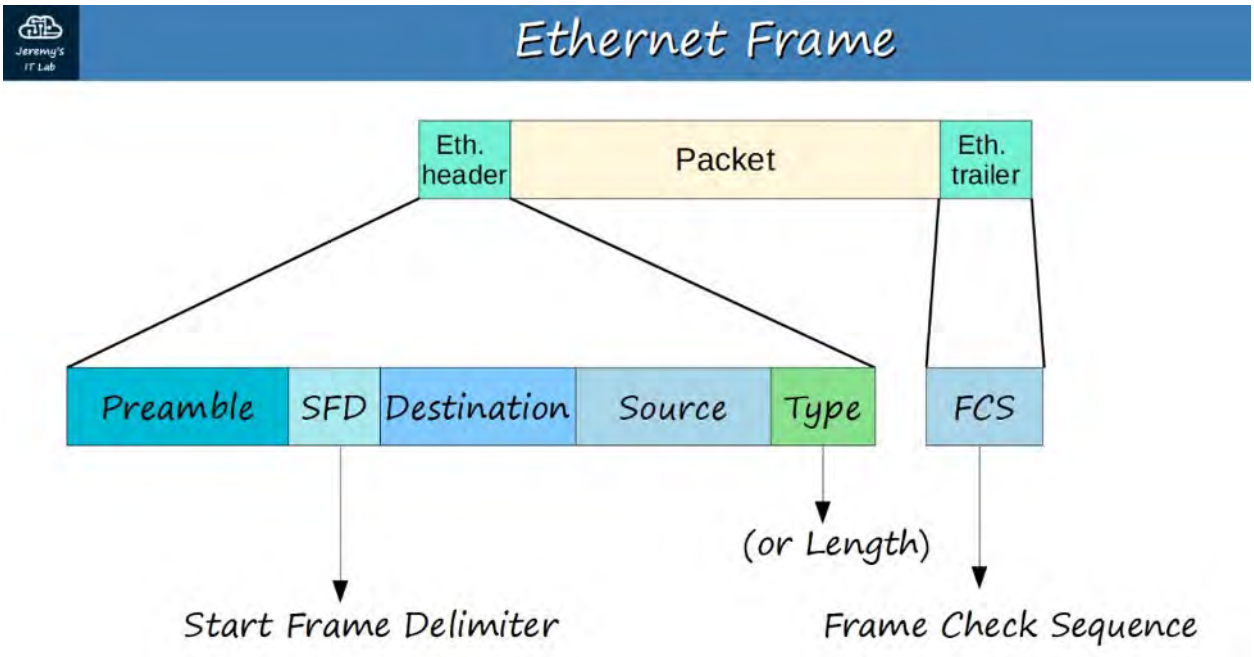
- Provides node-to-node connectivity and data transfer (for example, PC to switch, switch to router, router to router).
- Defines how data is formatted for transmission over a physical medium (for example, copper UTP cables)
- Detects and (possibly) corrects Physical Layer errors.
- Uses Layer 2 addressing, separate from Layer 3 addressing.
- Switches operate at Layer 2.

LAN's

- A LAN is a network contained in a relatively small area.
- Routers are used to connect separate LAN's



An ETHERNET FRAME looks like:



Ethernet Trailer --- PACKET --- Ethernet Header
The Ethernet Header contains 5 Fields:
Preamble -- SFD -- Destination -- Source -- Type 7 bytes -- 1 byte -- 6 bytes -- 6 bytes -- 2 bytes

---

PREAMBLE:

- Length: 7 bytes (56 bits)
- Alternating 1's and 0's
- 10101010 * 7x
- Allows devices to synchronize their receiver clocks

SFD : ‘Start Frame Delimiter’

- Length: 1 byte(8 bits)
- 10101011
- Marks end of the PREAMBLE and beginning of rest of frame.

DESTINATION AND SOURCE

- Layer 2 Address
- Indicates the devices sending / receiving the frame
- MAC = ’Media Access Control’
- = 6 byte (48-bit) address of the physical device

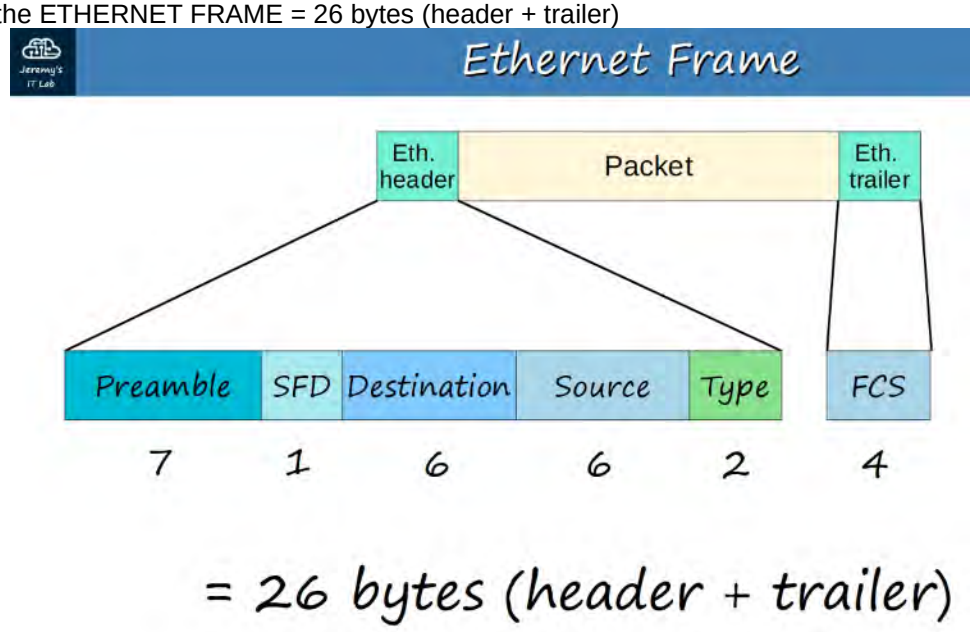
TYPE / LENGTH

- 2 bytes (16-bit) field
- A value of 1**500 or less** in this field indicates the LENGTH of the encapsulated packet (in bytes)
- A value of **1536 or greater** in this field indicates the TYPE of the encapsulated packet and length is determined via other methods.
- IPv4 = 0x0800 (hexadecimal) = 2048 in decimal
- IPv6 = 0x86DD (hexadecimal) = 34525 in decimal
- Layer 3 protocol used in the encapsulated Packet, which is almost always Internet Protocol (IP) version 4 or version 6.

The ETHERNET TRAILER contains:
FCS

- ‘FRAME CHECK SEQUENCE’
- 4 bytes (32 bits) in length
- Detects corrupted data by running a 'CRC' algorithm over the received data
- CRC = "Cyclic Redundancy Check"

Altogether the ETHERNET FRAME = 26 bytes (header + trailer)



MAC ADDRESS (48 bits long)

- 6-bytes (48-bits) physical address assigned to the device when it is made.
- AKA 'Burned-In Address' (BIA)
- Is globally unique

- 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
- Written as 12 hexadecimal characters

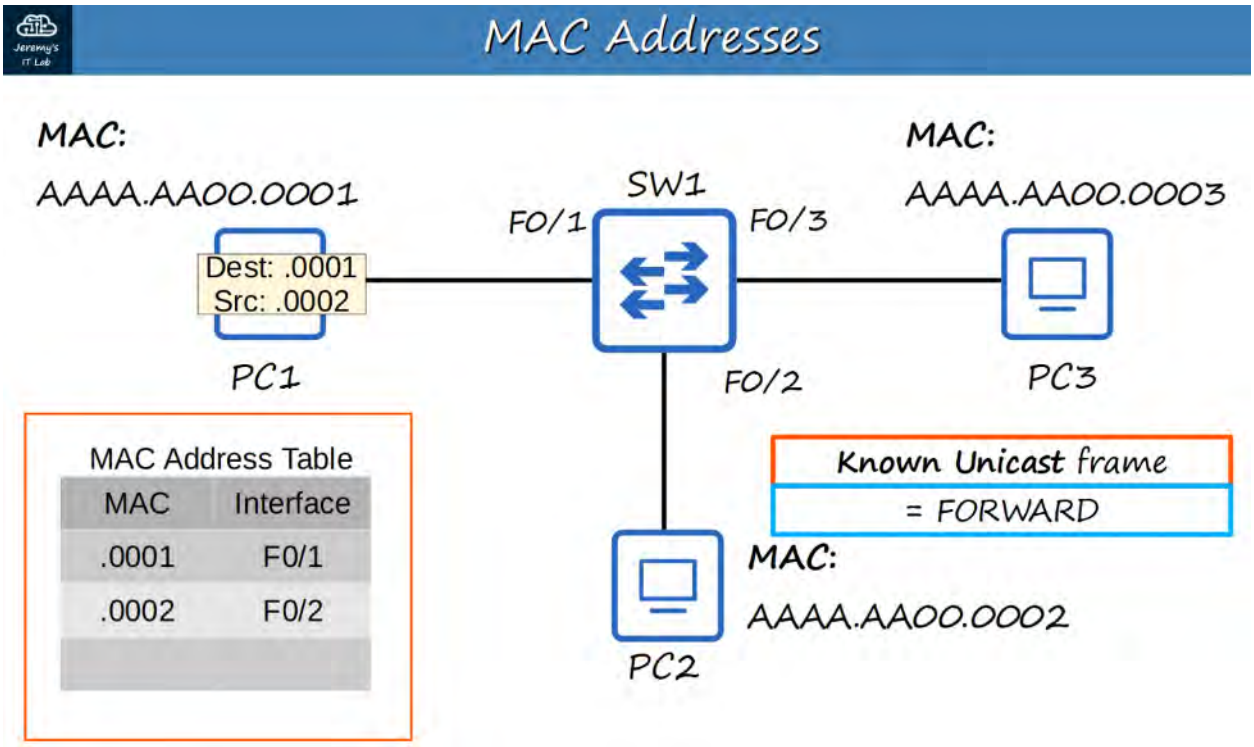
Example:
E8:BA:70 // 11:28:74 OUI // Unique Device ID
HEXADECIMAL

Hexadecimal

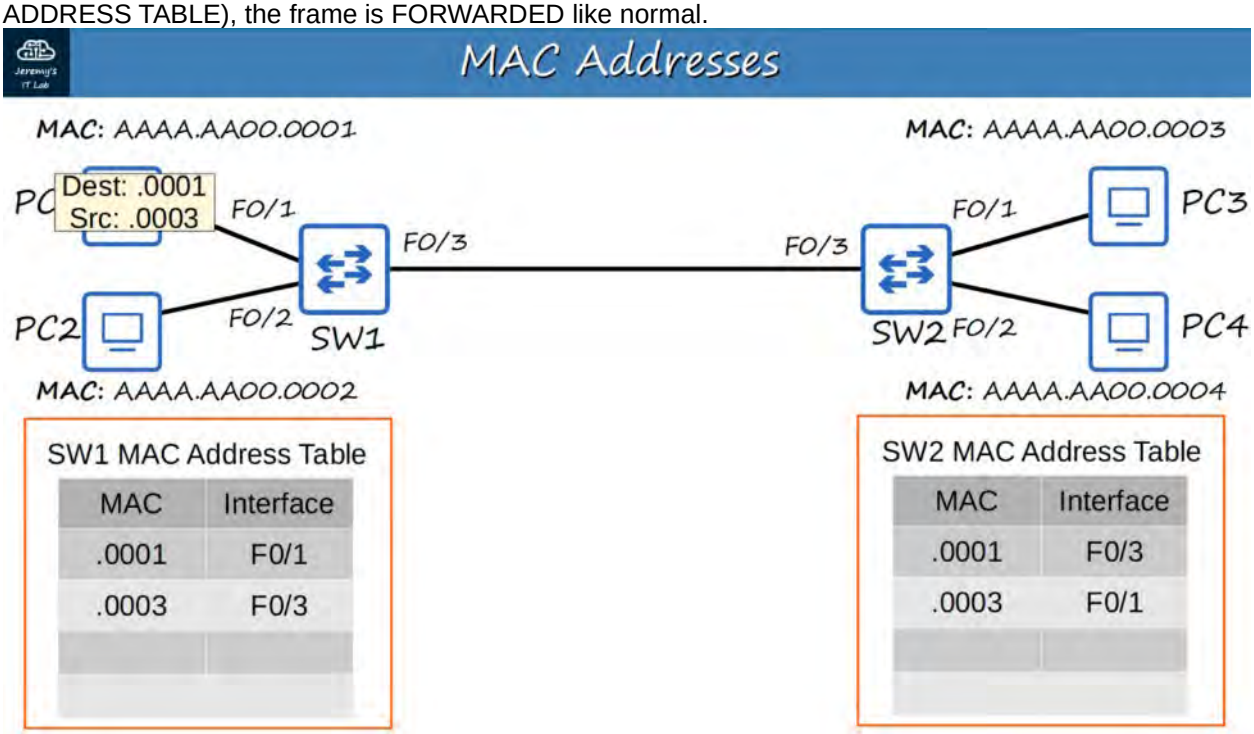
DEC.	HEX.	DEC.	HEX.	DEC.	HEX.	DEC.	HEX.
0	0	8	8	16	10	24	18
1	1	9	9	17	11	25	19
2	2	10	A	18	12	26	1A
3	3	11	B	19	13	27	1B
4	4	12	C	20	14	28	1C
5	5	13	D	21	15	29	1D
6	6	14	E	22	16	30	1E
7	7	15	F	23	17	31	1F

INTERFACE NAMES
F0/1, F0/2, F0/3... F stands for "Fast Ethernet" or 100 Mbps interfaces.

MAC ADDRESS TABLE
Each Switch stores a DYNAMICALLY LEARNED MAC ADDRESS TABLE, using the SOURCE MAC ADDRESS of frames it receives.



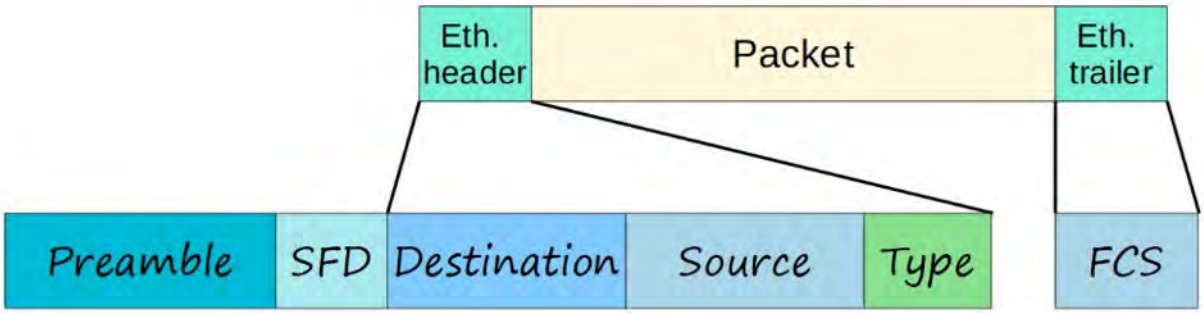
When a Switch doesn't know the DESTINATION MAC ADDRESS of a frame (UNKNOWN UNICAST FRAME), it is forced to FLOOD the frame - Forward the frame out of ALL it's interfaces, except the one it received the packet from.
When a KNOWN Unicast Frame is known (MAC Address is recognized by the entry in the MAC ADDRESS TABLE), the frame is FORWARDED like normal.



- Note: Dynamic MAC Addresses are removed from the MAC ADDRESS TABLE every 5 minutes of inactivity.

6. ETHERNET LAN SWITCHING : PART 2

An ETHERNET FRAME looks like:
Ethernet Header --- DATA (Packet) --- Ethernet Trailer



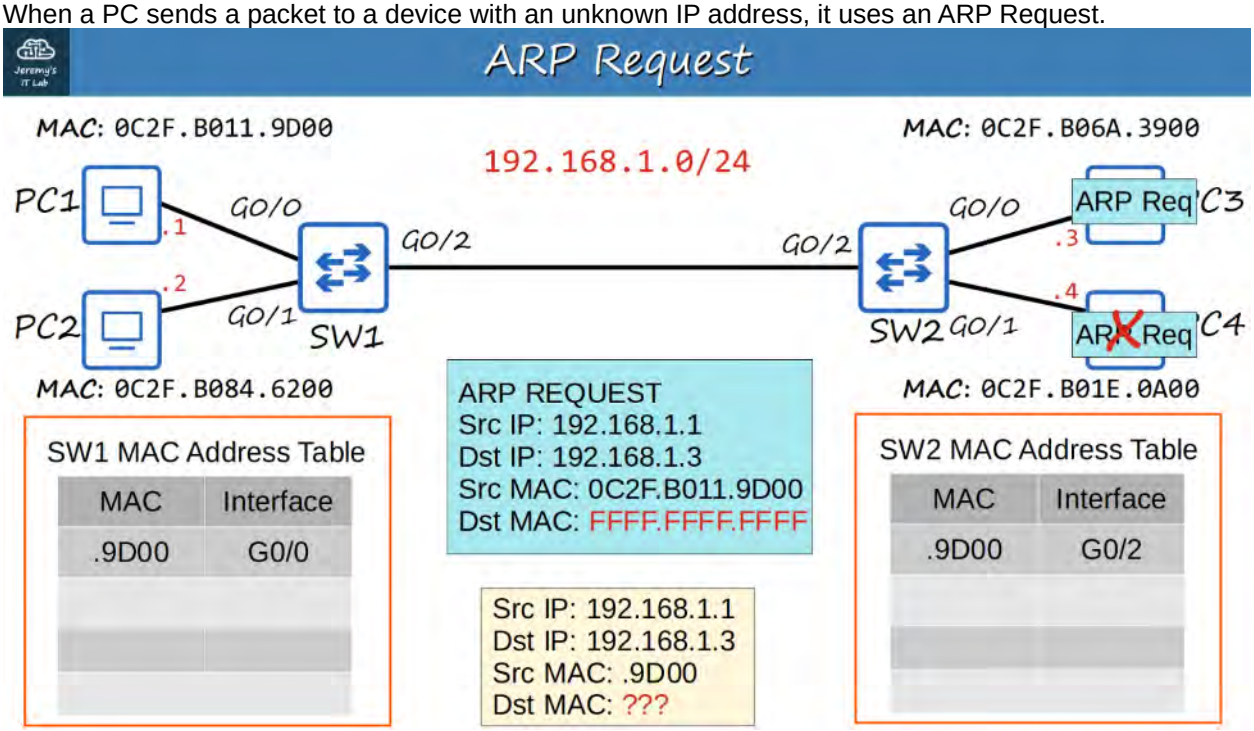
The Ethernet Header contains 5 Fields:
Preamble -- SFD -- Destination -- Source -- Type/Length 7 bytes -- 1 byte -- 6 bytes -- 6 bytes -- 2 bytes
Ethernet Trailer contains 1 Field:
FCS (Frame Check Sequence) = 4 bytes

- The PREAMBLE + SFD is not usually considered part of the ETHERNET HEADER.

THEREFORE the size of the ETHERNET HEADER + TRAILER is 18 bytes
(6 + 6 + 2 + 4 bytes for the FRAME CHECK SEQUENCE)

The MINIMUM size for an ETHERNET FRAME (Header + Payload [PACKET] + Trailer) is 64 BYTES.
64 BYTES - 18 BYTES (Header + Trailer size) = 46 BYTES
THEREFORE the MINIMUM DATA PAYLOAD (PACKET) size is 46 BYTES!
IF the PAYLOAD is LESS than 46 BYTES then PADDING BYTES are added (padding bytes are a series of 0's) until it equals to 46 BYTES.

When a PC sends a packet to a device with an unknown IP address, it uses an ARP Request.



- ARP stands for 'Address Resolution Protocol'.

- It is used to discover the Layer 2 address (MAC address) of a known Layer 3 address (IP address)
- Consists of two messages:
    - ARP REQUEST (Source message)
    - ARP REPLY (Destination message)
- ARP REQUEST is BROADCAST = sent to all hosts on network, except the one it received the request from.

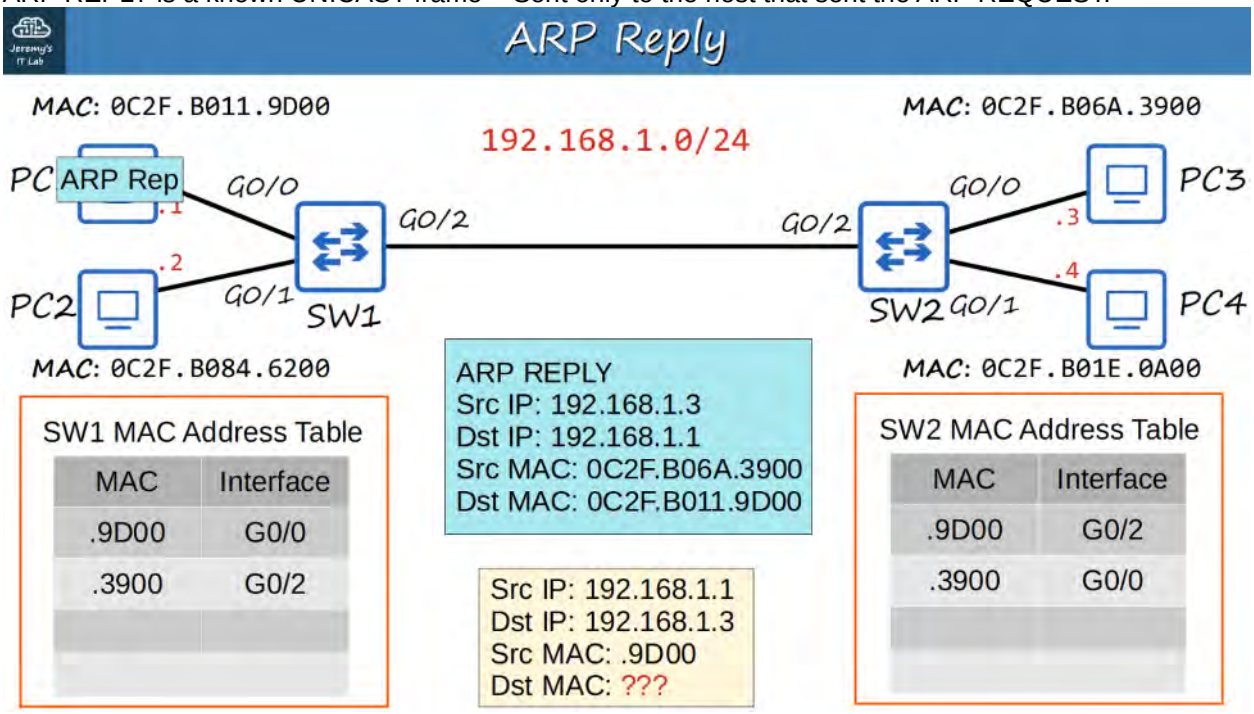
An ARP REQUEST frame has:

- Source IP Address
- Destination IP Address
- Source MAC address
- BROADCAST MAC Address - FFFF.FFFF.FFFF

An ARP REPLY frame has:

- Source IP Address
- Destination IP Address
- Source MAC address
- Destination MAC Address

ARP REPLY is a known UNICAST frame = Sent only to the host that sent the ARP REQUEST.



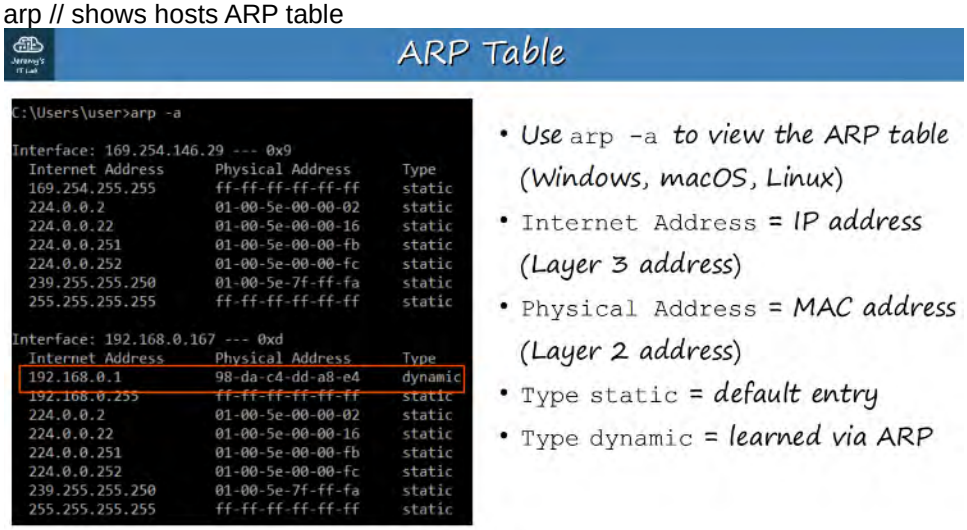
PING

- A network utility that is used to test reachability
- Measures round-trip time
- Uses two messages:
    - ICMP Echo REQUEST
    - ICMP Echo REPLY
- Is UNICAST
- Command to use ping:
    - ping

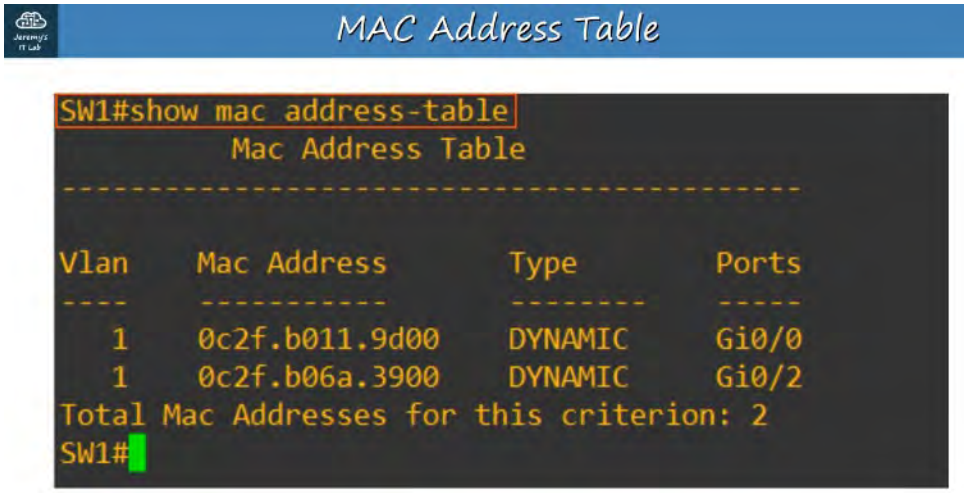
By Default, a CISCO IOS sends 5 ICMP requests/replies (Default size is 100-bytes)

- A period (.) is a failed ping
- An exclamation mark (!) is a successful ping

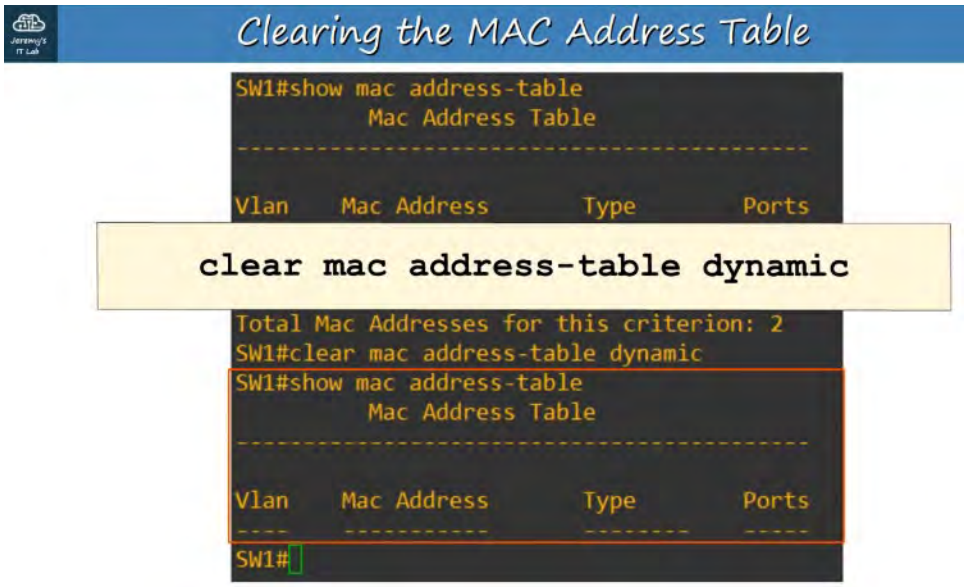
USEFUL CISCO IOS COMMANDS (from Privileged EXEC mode)
PC1# show arp // shows hosts ARP table



SW1#show mac address-table // show the switches MAC table



Will show:
Vlan --- MAC Address --- Type --- Ports(interfaces)
(Vlan = Virtual Local Area Network)



SW1# clear mac address-table dynamic
// clears the entire switches MAC table. // IF the optional MAC address is used, it will clear the SPECFIC MAC address.
SW1 #clear mac address-table dynamic interface
// clears the MAC table entry of the Switch by it's **INTERFACE n**ame.

7. IPv4 ADDRESSING : PART 1

OSI MODEL - NETWORK LAYER (Layer 3)

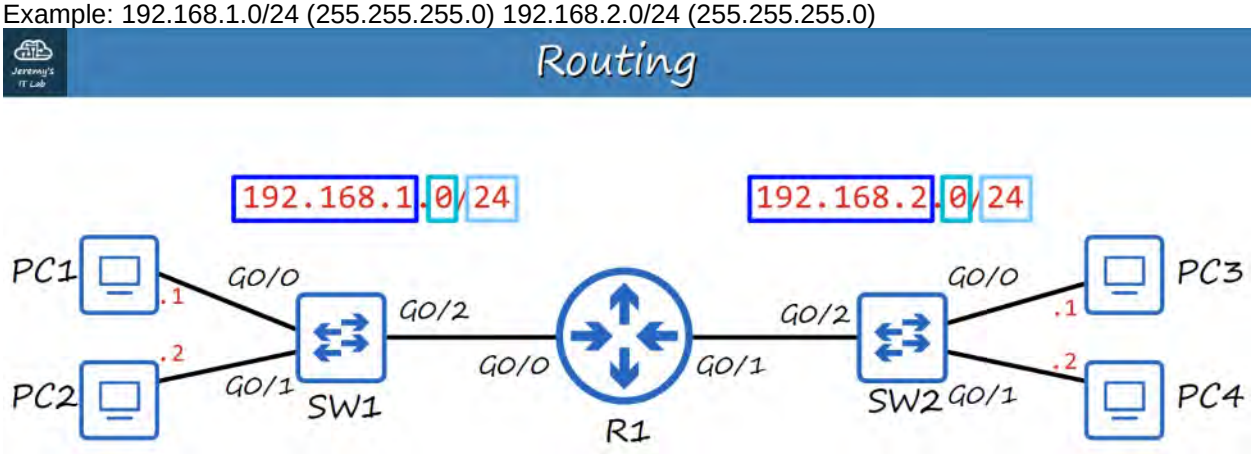
- Provides connectivity between end hosts on DIFFERENT networks (ie: outside of the LAN)
- Provides logical addressing (IP addresses)
- Provides path selection between SOURCE and DESTINATION
- ROUTERS operate at LAYER 3

ROUTING

SWITCHES (Layer 2 Devices) do no separate different networks. They connect and EXPAND networks within the same LAN.

By adding a ROUTER, however, between two SWITCHES, you create a SPLIT in the network; each with it's own network IP address.

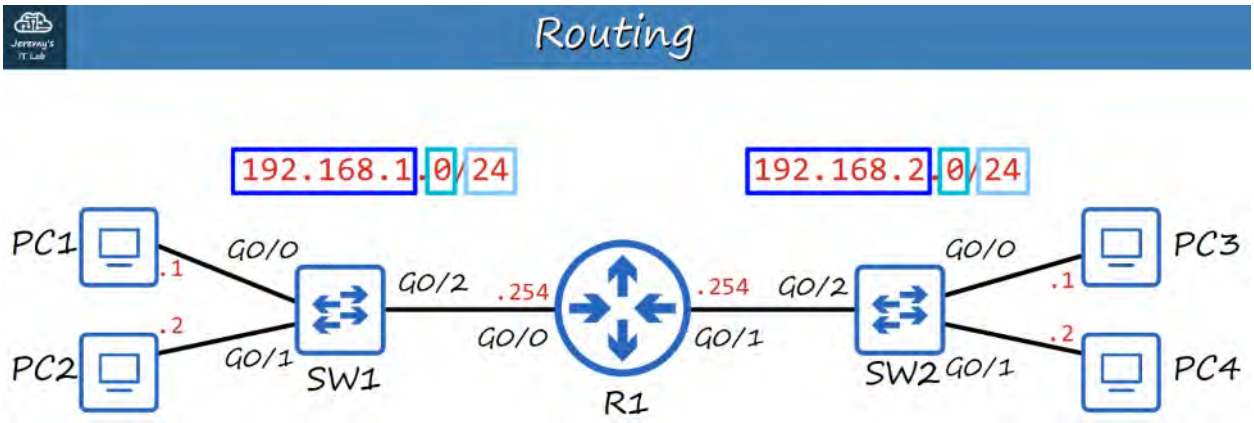
Example: 192.168.1.0/24 (255.255.255.0) 192.168.2.0/24 (255.255.255.0)



ROUTERS have unique IP Addresses for EACH of their interface connections, depending on their location.

The IP Address for the ROUTER's G0/0 Interface is: 192.168.1.254/24

The IP Address for the ROUTER's G0/1 Interface is: 192.168.2.254/24



The IP Address depends on network address of the LAN it is connects to.

The NETWORK portion of given IP Address will be the same for all HOSTS on a given LAN.

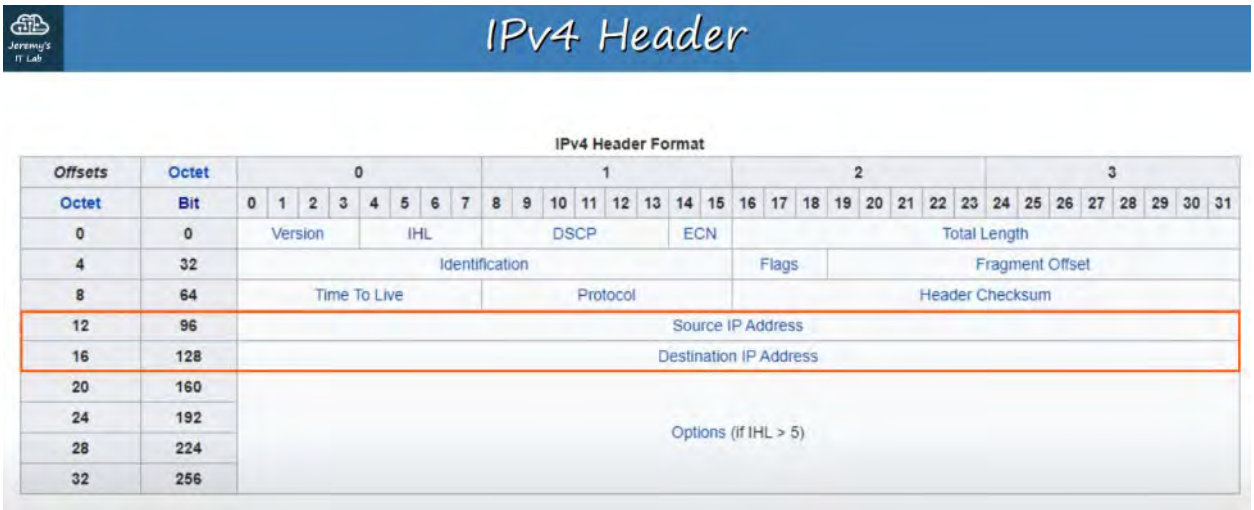
Example:

192.168.1.100 192.168.1.105 192.168.1.205

All of these addresses are on the SAME Network because the NETWORK PORTION of their IP Address is the same (192.168.1) while the HOST part (100,105,205) is UNIQUE!

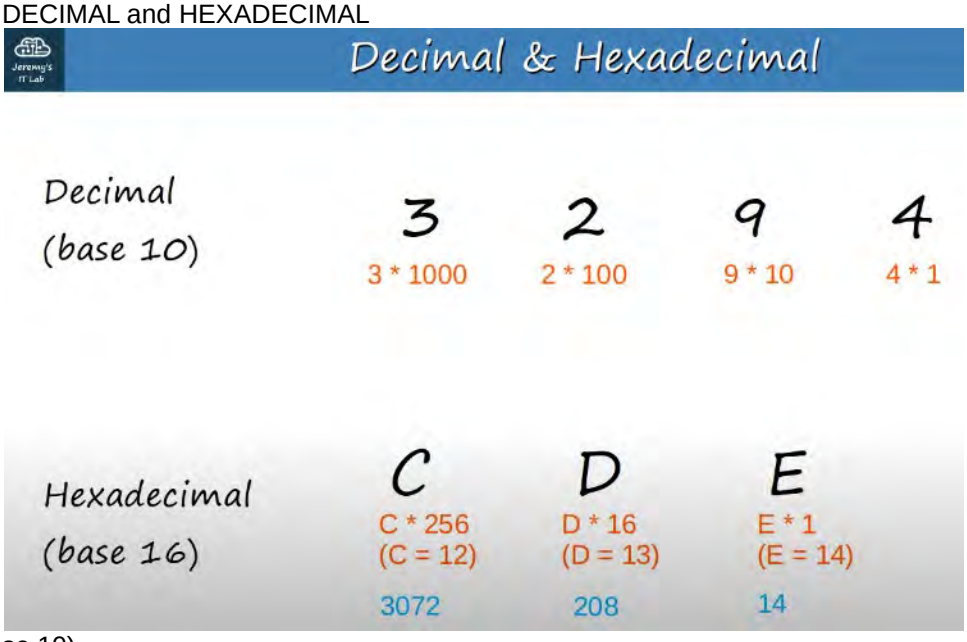
When a BROADCAST message hits a ROUTER, it does NOT continue onward. It stays within the LOCAL LAN (Switch/Hosts).

IPv4 HEADER



IP (or Internet Protocol) is the primary Layer 3 protocol in use today. Version 4 is the version in use in most networks.
IPv4 Headers contain MORE fields than the ETHERNET header.
IPv4 Headers contain a SOURCE IP Address and DESTINATION IP Address field.
This FIELD is 32-bits(4-bytes) in length (0-31)
192.168.1.254 (each decimal number represents 8 bits)
Translated to Binary:
11000000 . 10101000 . 00000001 . 11111110
EACH of these 8 bit groups are referred to as an OCTET
Since Binary is difficult to read for people, we use the Dotted Decimal format.

REVIEW of DECIMAL and HEXADECIMAL



Decimal (base 10)
Ex: 3294 = (3 * 1000) + (2 * 100) + (9 * 10) + (4 * 1)
Hexadecimal (base 16)
Ex: 3294, would be CDE
C (C * 256 / 12 * 256 = 3072) // 256ths position
D (D * 16 / D=13 so 16*13 = 208) // 16ths position
E (E * 1 / E = 14) // 1s position
Adding these up, we get 3294

So, how do we convert a BINARY NUMBER to a DECIMAL NUMBER? The same way we convert to Hexadecimal.
10001111
So:
1 * 128 = 128
1 * 8 = 8
1 * 4 = 4
1 * 2 = 2
1 * 1 = 1
Add them all up : 128 + 8 + 4 + 2 + 1 = 143
The answer is 143.

Another example:
01110110
1 * 64 = 64
1 * 32 = 32
1 * 16 = 16
1 * 4 = 4
1 * 2 = 2
Add them all up: 64 + 32 + 16 + 4 + 2 = 118
The answer is 118.

Another example:
11101100
1 * 128 = 128
1 * 64 = 64
1 * 32 = 32
1 * 8 = 8
1 * 4 = 4
Add them all up: 128 + 64 + 32 + 8 + 4 = 236
The answer is 236.

So, how do we convert a DECIMAL NUMBER to a BINARY NUMBER?
Take the number 221.
We can take that number and start subtracting it from LEFT to RIGHT of our Binary slots.
221
221 - 128 = 93 so we place a 1 in the "128" slot
10000000
93 - 64 = 29 so we place another 1 in the "64" slot

29 - 32 isn't possible so we place a 0 in the "32" slot

29 - 16 = 13 so we place a 1 in the "16" slot

13 - 8 = 5 so we place a 1 in the "8" slot

5 - 4 = 1 so we place a 1 in the "4" slot

1 - 2 isn't possible so we put a 0 in the "2" slot

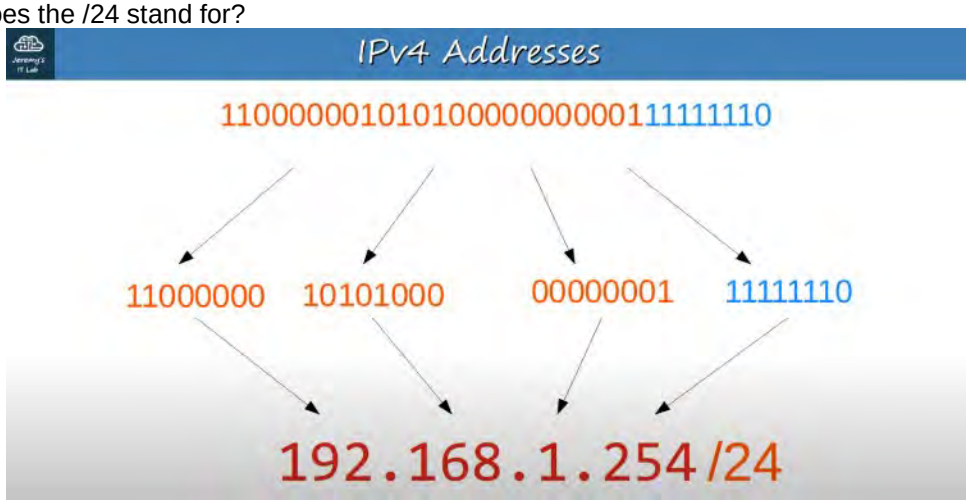
1 - 1 is possible so we put a 1 in the "1" slot
This, then, allows us to the write out the BINARY number for 221.
It is : 11011101

Another example: 127

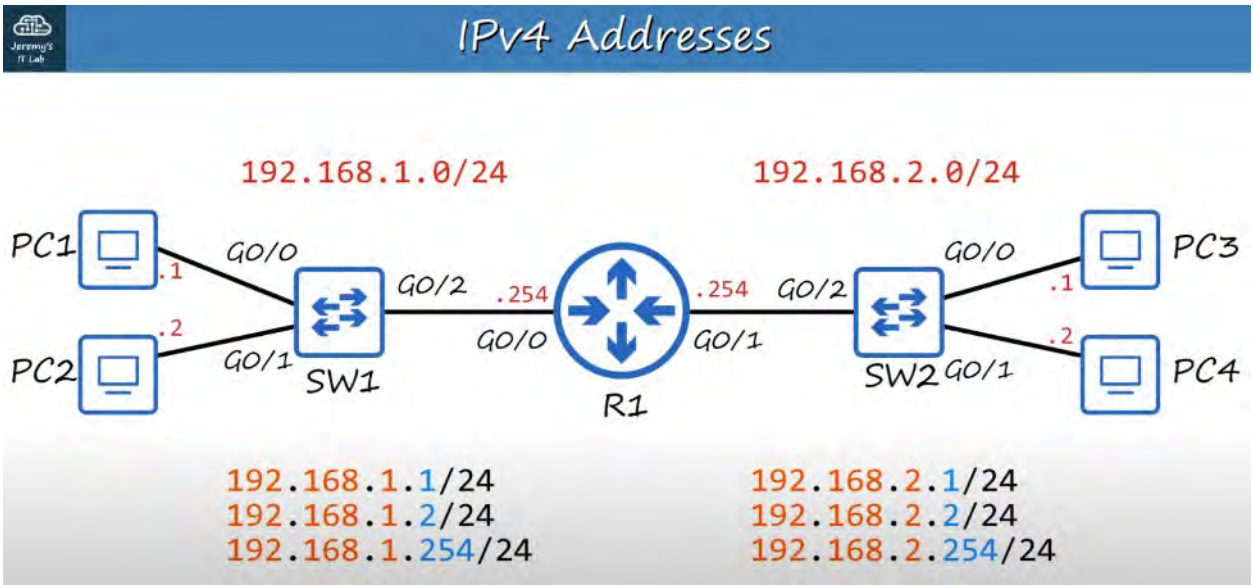
127 - 128 is not possible so 0 in "128"
127 - 64 is possible so 1 in "64"
63 - 32 is possible so 1 in "32"
31 - 16 is possible so 1 in "16"
15 - 8 is possible so 1 in "8"
7 - 4 is possible so 1 in "4"
3 - 2 is possible so 1 in "2"
1 is possible so 1 in "1"
So 127, in BINARY, is 0111 1111

Another example: 207
Alternatively, you can subtract the number from '255' (which is 1111111). The remainder, then, can be used to "find" where the 0's are in the binary number.
255 - 207 = 48 so ...
1 1 0 0 1 1 1 1 (32 + 16 = 48)
11001111 is the correct answer.

IPv4 ADDRESSES
So we now know that IP Addresses are the Dotted Decimal conversion of a series of BINARY NUMBERS (broken up into 4 OCTETS) like so:
192.168.1.254/24
But what does the /24 stand for?



It means the FIRST 24 BITS of this address represent the NETWORK portion of the address.
192.168.1 is the NETWORK PORTION (the first 3 OCTETS)
.254 is the HOST PORTION (the last OCTET)



CONVERT this BINARY number into an IPv4 Address:
10011010010011100110111100100000
10011010 . 01001110 . 01101111 . 00100000
Octets:

1. 128 + 16 + 8 + 2 = 154
2. 64 + 8 + 4 + 2 = 78
3. 64 + 32 + 8 + 4 + 2 + 1 = 111
4. 32

The IPv4 address is: 154.78.111.32/16
154.78 is the NETWORK PORTION 111.32 is the HOST PORTION
Another Example:
00001100100000001111101100010111
00001100 . 10000000 . 11111011 . 00010111
Octets:

1. 8 + 4 = 12
2. 128
3. 255 - 4 = 251
4. 16 + 4 + 2 + 1 = 23

The IPv4 address is: 12.128.251.23/8
12 is the NETWORK PORTION 128.251.23 is the HOST PORTION

IPv4 ADDRESS CLASSES
IPv4 ADDRESSES are split up into 5 different 'classes'. The class of an IPv4 is determined by the FIRST OCTET of the address.
CLASS FIRST OCTET FIRST OCTET NUMBERIC RANGE
A 0xxxxxxx 0-126 + 127 'loopback' B 10xxxxxx 128-191 C 110xxxxx 192-223 D 1110xxxx 224-239 E 1111xxxx 240-255
From the above chart, if the FIRST OCTECT STARTS with 0, the numeric RANGE of possible first DOTTED DECIMAL is between 0-127.
The CLASSES we will be focusing on are CLASS A to CLASS C.

IPv4 Address Classes

Class	Leading bits	Size of *network number* bit field	Size of *rest* bit field	Number of networks	Addresses per network
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})
Class C	110	24	8	2,097,152 (2^{21})	256 (2^8)

D CLASS are reserved for 'MULTICAST' ADDRESSES
E CLASS are reserved for 'EXPERIMENTAL' ADDRESSES

A CLASS USUALLY have a range of 1-126? WHY?
Because 127 is usually reserved for 'loopback addresses'
127.0.0.0 to 127.255.255.255 are used to test the network.

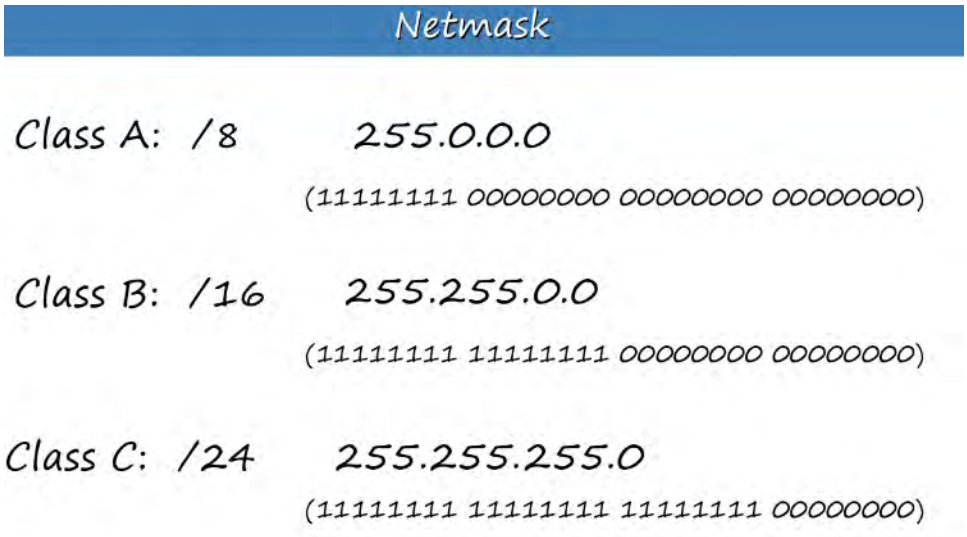
- Used to test the 'Network stack' (OSI & TCP/IP model) on the local device.

IPv4 Address Classes

Class	First octet	First octet numeric range	Prefix Length
A	0xxxxxxx	0-127	/8
B	10xxxxxx	128-191	/16
C	110xxxxx	192-223	/24

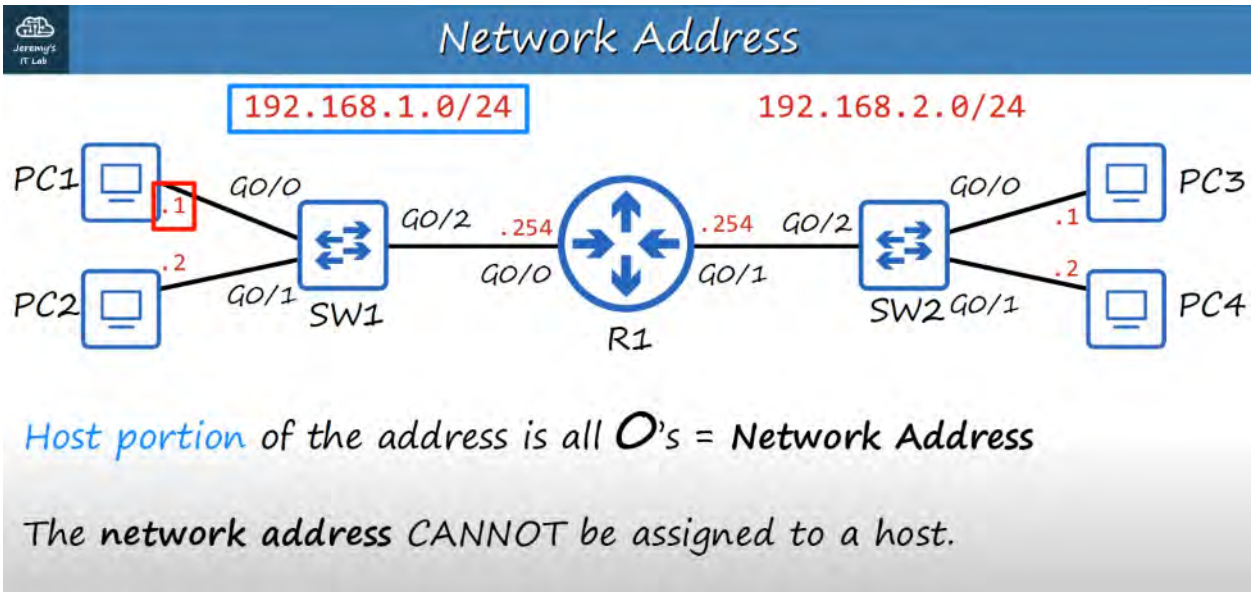
The PREFIX LENGTH is the LENGTH of the NETWORK PORTION of the Address.
From the examples above:
12.128.251.23/8 is a CLASS A Address 154.78.111.32/16 is a CLASS B Address 192.168.1.254/24 is a CLASS C Address
Because the NETWORK portion of CLASS A is so short, it means there are a LOT more potential Hosts.
Because the NETWORK portion of CLASS C is so long, it means fewer potential Hosts.

NETMASK

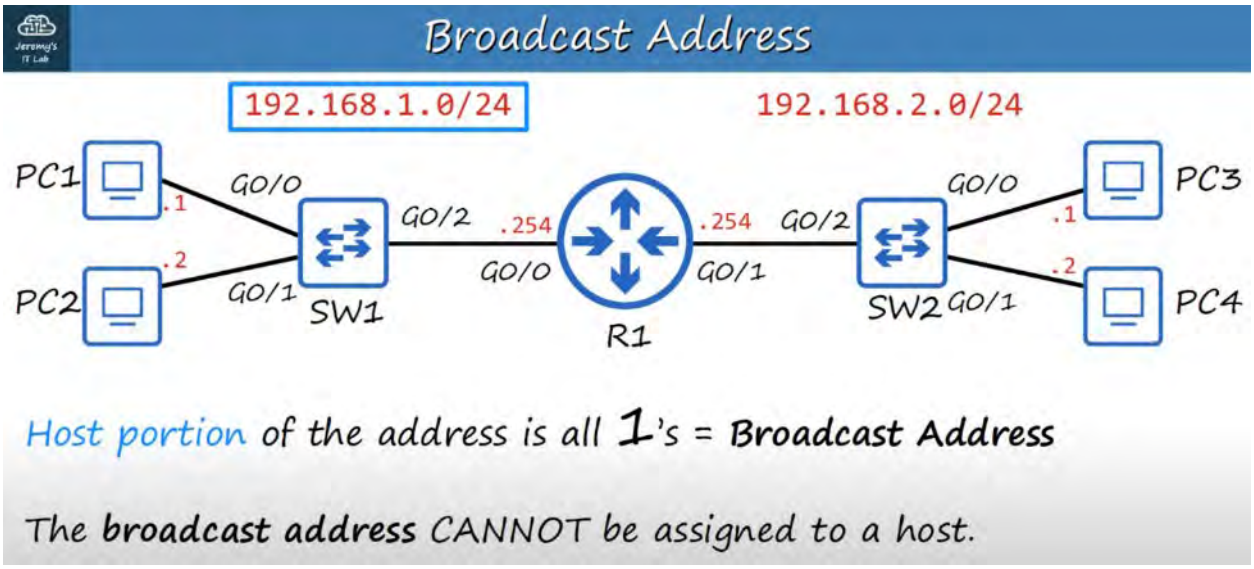


A NETMASK is written like a Dotted Decimal IP Address
CLASS A: /8 = 255.0.0.0
CLASS B: / 16 = 255.255.0.0
CLASS C: /24 = 255.255.255.0

NETWORK ADDRESSES



If the HOST PORTION of an IP ADDRESS is ALL 0's, it means it is the NETWORK ADDRESS = the identifier of the network itself.
Example: 192.168.1.0/24 = THIS is a NETWORK ADDRESS.
A NETWORK ADDRESS cannot be assigned to a HOST. A NETWORK ADDRESS is the FIRST ADDRESS.



If the HOST PORTION of an IP ADDRESS is ALL 1's, it means it is the BROADCAST ADDRESS for the network.
A BROADCAST ADDRESS cannot be assigned to a HOST.
DESTINATION IP : 192.168.1.255 (Broadcast IP address) DESTINATION MAC : FFFF.FFFF.FFFF (Broadcast MAC address)
Because of the two 'reserved' addresses, the range of USABLE HOST ADDRESSES is 1 to 254.

8. IPv4 ADDRESSING : PART 2

MAXIMUM HOSTS PER NETWORK
Let's take a Class C Network:
192.168.1.0/24
(gives a range of 0 ---> 255)
Said another way, the HOST portion (the .0) is equal to 8 bits so...
Host portion = 8 bits = 2^8 = 256
HOWEVER, since the Network Address (Network ID)
192.168.1.0 is Reserved
AND
192.168.1.255 (BROADCAST ADDRESS) is ALSO reserved.
The MAXIMUM Hosts per Network = 2^8-2 = 254 hosts

What about a Class B Network ?
172.16.0.0/16 ----> 172.16.255.255/16
Host portion = 16 bits = 2^16 = 65,536
Maximum hosts per network = 2^16-2 = 65,534 hosts

What about a Class A Network ?
10.0.0.0/8 ----------> 10.255.255.255/8
Host portion = 24 bits = 2^24 = 16,777,216
Maximum hosts per network = 2^24-2 = 16,777,214 hosts

THEREFORE:
The formula for calculating the number of HOSTS on a network is:
2 ^ N - 2 (2 to the power of N - 2)
where N = number of HOST bits

FIRST / LAST USABLE ADDRESSES
Class C Network
192.168.1.0/24 (NETWORK ADDRESS)
Add 1 so the Host Portion = 00000001
192.168.1.1/24 = FIRST USABLE ADDRESS

192.168.1.255/24 (BROADCAST ADDRESS)
Subtract 1 from the BROADCAST ADDRESS = 11111110
192.168.1.254/24 = LAST USABLE ADDRESS

Class B Network
172.16.0.0/16 (NETWORK ADDRESS)
Add 1 to Host portion so 0000 0000 0000 0001
172.16.0.1/16 is the FIRST USABLE ADDRESS

172.16.255.255/16 (BROADCAST ADDRESS)
Subtract 1 to Broadcast Address so 1111 1111 1111 1110
172.16.255.254/16 is the LAST USABLE ADDRESS

Class A Network
10.0.0.0/8 (NETWORK ADDRESS)
Add 1 to Host portion so 00000000 00000000 00000001
10.0.0.1/8 is the FIRST USABLE ADDRESS

10.255.255.255/8 (BROADCAST ADDRESS)
Subtract 1 to Broadcast Address so 1111 1111 1111 1110
10.255.255.254/16 is the LAST USABLE ADDRESS

CISCO CLI DEVICE CONFIGURATION
R1> enable R1# show ip interface brief
Lists the Interfaces, IP Addresses, Method, Status, and Protocol.
Interfaces:

- What port interfaces are available/connected

IP Addresses

- Self explanatory. What IP Address is assigned.

Method

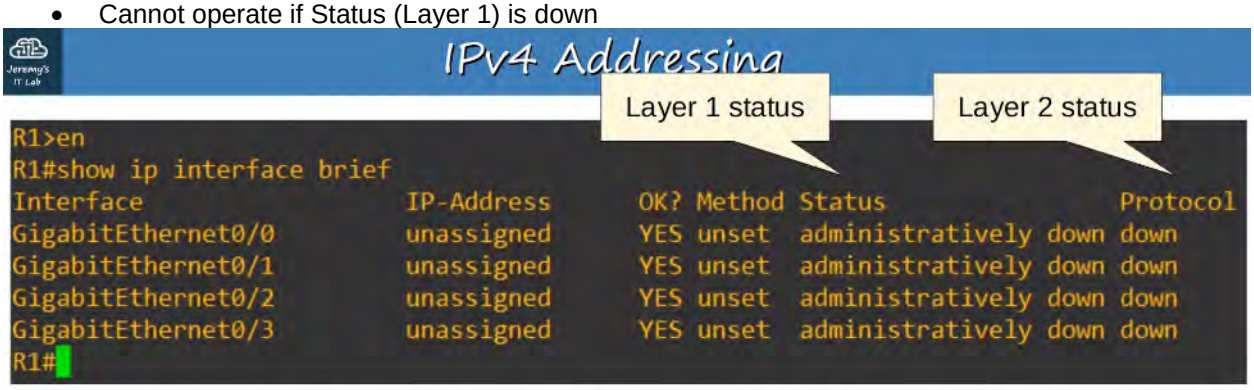
- What method was the IP address assigned?

Status (Layer 1 Status)

- Current status of interface
- 'administratively down' = Interface has been disabled with the 'shutdown' command

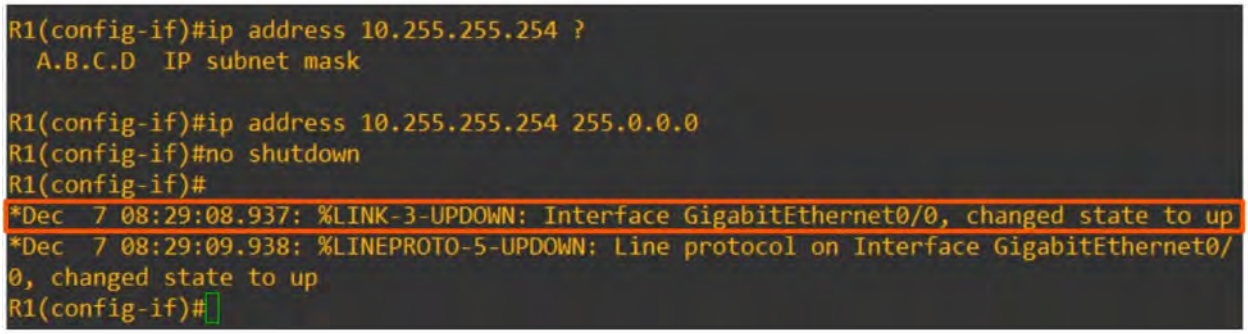
Administratively down is the DEFAULT status of Cisco Router interfaces.
Cisco Switch interfaces are NOT administratively down by DEFAULT.
Protocol (Layer 2 Status)

- Cannot operate if Status (Layer 1) is down



- `administratively down`: Interface has been disabled with the 'shutdown' command.
- This is the default `Status` of Cisco router interfaces.
- Cisco switch interfaces are NOT `administratively down` by default.

// configure terminal cmd
R1# conf t
// This enters interface configuration mode
R1(config)# interface gigabitethernet 0/0
This can be shortened to 'g0/0' like they are listed in physical network maps.

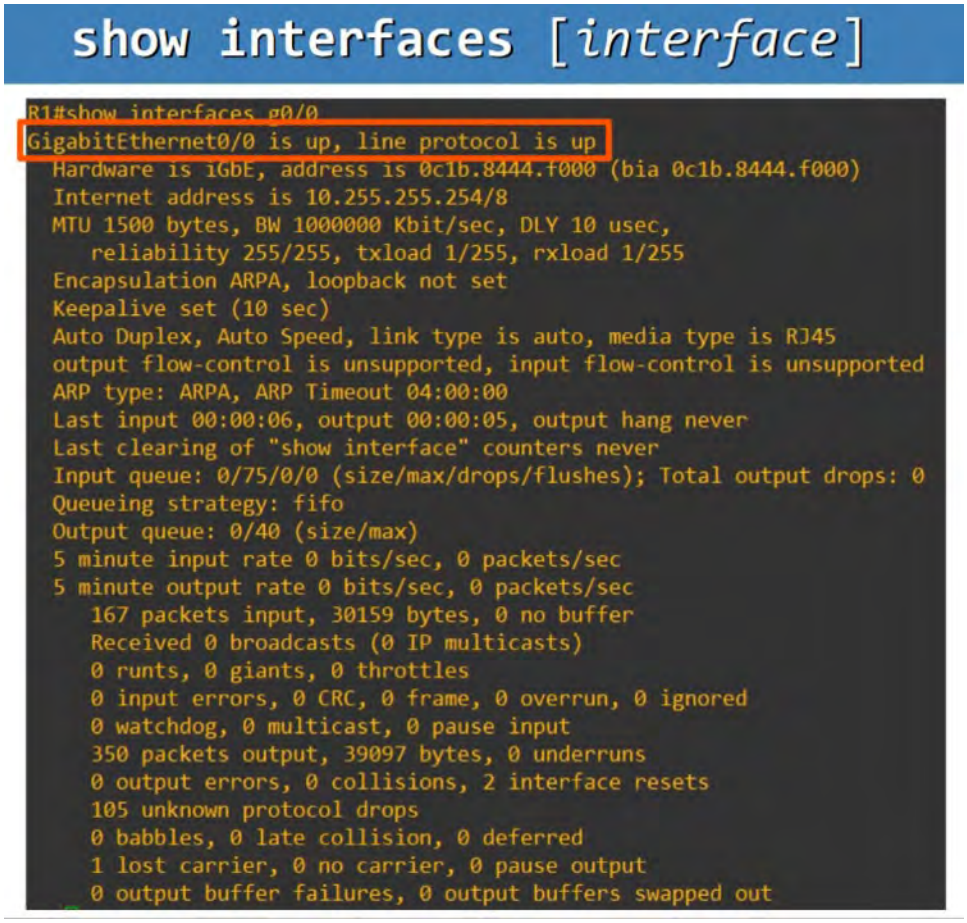


// This sets the IP ADDRESS and SUBNET MASK of device
R1(config-if) #ip address 10.255.255.254 255.0.0.0

// This enables the device
R1(config-if) #no shutdown

Two messages should appear showing the state has changed to 'up' (Status). Second message should show line protocol is now 'up' (Protocol).
// 'do' allows you to run a Privileged EXEC command from outside the mode.
R1(config-if) #do show ip interface brief
Good to confirm that the device/interface you have configured is up and running.

More 'show' CLI Commands



'show interfaces '

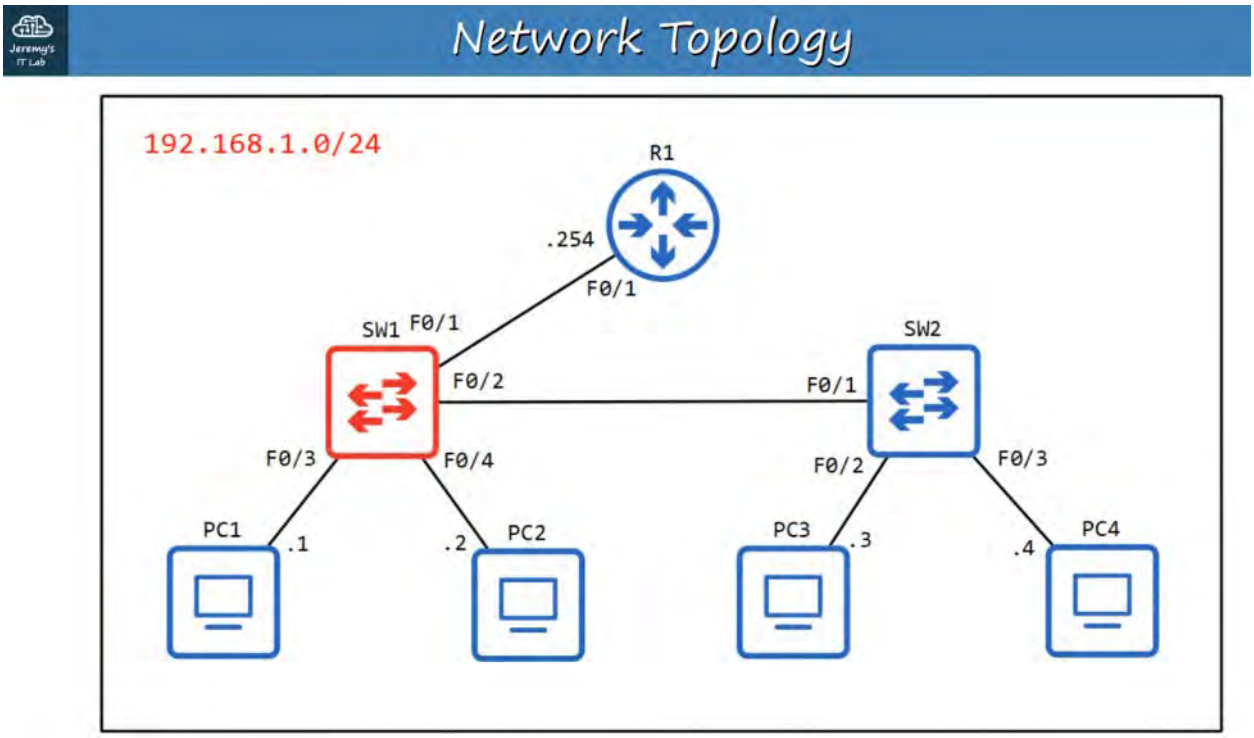
- Shows Layer 1 and Layer 2 information about the interface and some Layer 3.
- Shows MAC Address (or BIA address)
- IP Address
- ... and so much more

'show interfaces description'

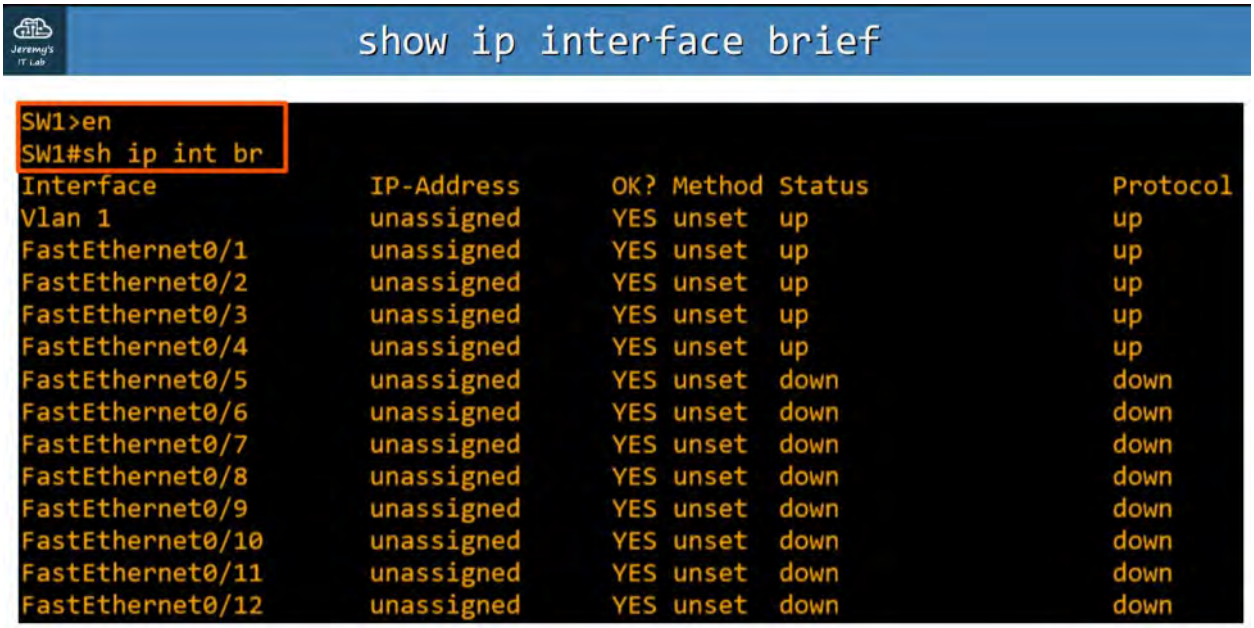
- Allows you to add descriptions for interfaces.

Example:
// Configure mode for interface Gigabyte Interface 0/0
R1(config) #int g0/0
R1(config) #description ## to SW1 ##
This sets the 'Description' column to display:
Interface Description
Gi0/0 ## to SW1 ##

9. SWITCH INTERFACES



CISCO CLI for SWITCHES



// enter Privileged EXEC mode
SW1>enable
// Show all interfaces of Switch 1.
SW# show ip interface brief
This will show the interfaces currently on Switch 1. It has the same information structure as Cisco Routers.
Notice the Status (Layer 2) and Protocol (Layer 1) columns are showing "up/up".
Unlike ROUTERS, SWITCHES do no DEFAULT to 'administrative down/down'(shutdown).

Unconnected devices will show as "down" and "down" (not connected to another device)

show interfaces status

```
SW1#show interfaces status
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1                        connected    1          a-full  a-100 10/100BaseTX
Fa0/2                        connected    trunk      a-full  a-100 10/100BaseTX
Fa0/3                        connected    1          a-full  a-100 10/100BaseTX
Fa0/4                        connected    1          a-full  a-100 10/100BaseTX
Fa0/5                        notconnect   1            auto   auto 10/100BaseTX
Fa0/6                        notconnect   1            auto   auto 10/100BaseTX
Fa0/7                        notconnect   1            auto   auto 10/100BaseTX
Fa0/8                        notconnect   1            auto   auto 10/100BaseTX
Fa0/9                        notconnect   1            auto   auto 10/100BaseTX
Fa0/10                       notconnect   1            auto   auto 10/100BaseTX
Fa0/11                       notconnect   1            auto   auto 10/100BaseTX
Fa0/12                       notconnect   1            auto   auto 10/100BaseTX
```

// Show the status of all interfaces on SW1
SW1#show interfaces status
This will list:

- Ports
- Name (which is description)
- Status (connection status)
- Vlan (can be used to divide up LANs) - Vlan 1 is the default.
- Duplex (can the connection send/receive at same time?) - Auto is default
- Speed (speed in bps) - Auto is default
- Type (what medium is being used, speed of interface)

Configuring interface speed and duplex

```
SW1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW1(config)#int f0/1
SW1(config-if)#speed ?
10                  Force 10 Mbps operation
100                 Force 100 Mbps operation
auto                Enable AUTO speed configuration
SW1(config-if)#speed 100
SW1(config-if)#duplex ?
auto                Enable AUTO duplex configuration
full                Force full duplex operation
half                Force half-duplex operation
SW1(config-if)#duplex full
SW1(config-if)#description ## to R1 ##
```

Configuring interface speed and duplex

```
SW1#sh int status
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1     ## to R1 ##        connected    1            full    100 10/100BaseTX
Fa0/2                        connected    trunk      a-full  a-100 10/100BaseTX
Fa0/3                        connected    1          a-full  a-100 10/100BaseTX
Fa0/4                        connected    1          a-full  a-100 10/100BaseTX
Fa0/5                        notconnect   1            auto   auto 10/100BaseTX
Fa0/6                        notconnect   1            auto   auto 10/100BaseTX
Fa0/7                        notconnect   1            auto   auto 10/100BaseTX
Fa0/8                        notconnect   1            auto   auto 10/100BaseTX
Fa0/9                        notconnect   1            auto   auto 10/100BaseTX
Fa0/10                       notconnect   1            auto   auto 10/100BaseTX
Fa0/11                       notconnect   1            auto   auto 10/100BaseTX
Fa0/12                       notconnect   1            auto   auto 10/100BaseTX
```

INTERFACE RANGE
Unused Interfaces can pose a security risk so it's a good idea to deactivate them.
However, if you have 28+ interfaces not in use, do you have to do them one at a time?
Answer: No! There is a command to apply configurations to a range of interfaces.
Inside Global Config Mode (config t):

interface range

```
SW1(config)#interface range f0/5 - 12
SW1(config-if-range)#description ## not in use ##
SW1(config-if-range)#shutdown
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/5, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/6, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/7, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/8, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/9, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/10, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/11, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/12, changed state to administratively down
SW1(config-if-range)#
```

SW1(config)#interface range f0/5 - 12 // Choose all interfaces from 0/5 to 0/12
SW1(config-if-range)#description ## not in use ##
SW1(config-if-range)#shutdown
<< this will list all the interfaces being set to administratively down >>
Confirm with 'show interface status' in Privileged EXEC mode or if in CONFIG mode, use 'do show interface status'

Configuring switch interfaces

```
SW1(config-if-range)#do sh int status
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1     ## to R1 ##        connected    1            full    100 10/100BaseTX
Fa0/2     ## to SW2 ##       connected    trunk      a-full  a-100 10/100BaseTX
Fa0/3     ## to end hosts ## connected    1          a-full  a-100 10/100BaseTX
Fa0/4     ## to end hosts ## connected    1          a-full  a-100 10/100BaseTX
Fa0/5     ## not in use ##   disabled     1            auto   auto 10/100BaseTX
Fa0/6     ## not in use ##   disabled     1            auto   auto 10/100BaseTX
Fa0/7     ## not in use ##   disabled     1            auto   auto 10/100BaseTX
Fa0/8     ## not in use ##   disabled     1            auto   auto 10/100BaseTX
Fa0/9     ## not in use ##   disabled     1            auto   auto 10/100BaseTX
Fa0/10    ## not in use ##   disabled     1            auto   auto 10/100BaseTX
Fa0/11    ## not in use ##   disabled     1            auto   auto 10/100BaseTX
Fa0/12    ## not in use ##   disabled     1            auto   auto 10/100BaseTX
```

FULL / HALF DUPLEX
HALF DUPLEX:

- Device cannot send / receive data at the same time. If it is receiving a frame, it must wait before sending a frame.

FULL DUPLEX:

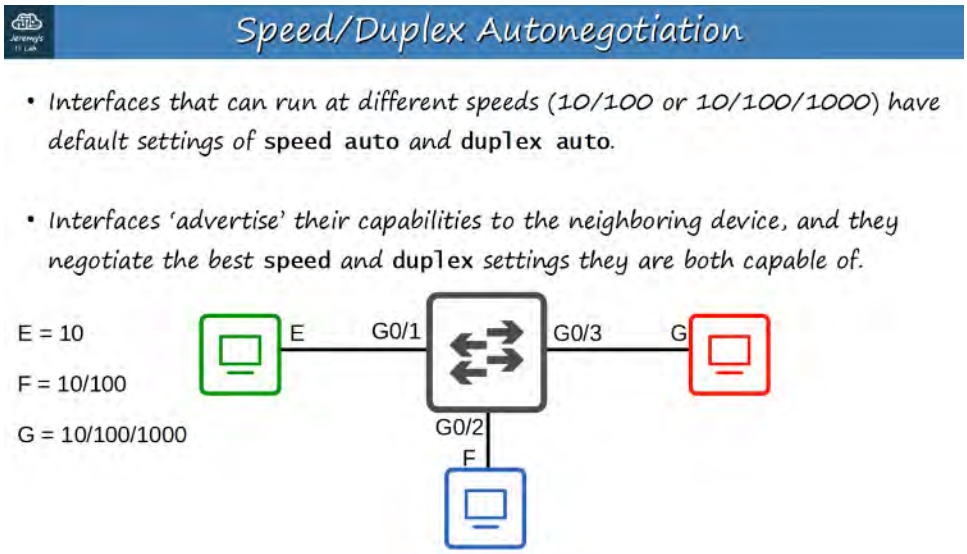
- Device CAN send / receive data at the same time. It does NOT have to wait.

MOST modern SWITCHES support FULL DUPLEX.

WHERE is HALF DUPLEX used? Almost nowhere.
In the past, LAN HUBS used HALF DUPLEX.
When multiple packets were received by the HUB, the HUB would simple FLOOD the connections with frame data, causing a COLLISION (on the interface), and hosts would not receive the frame intact.
All devices connected to a HUB are called a COLLISION DOMAIN.
To DEAL with COLLISIONS, Ethernet devices use a mechanism called CSMA/CD.
CSMA/CD = CARRIER SENSE MULTIPLE ACCESS with COLLISION DETECTION.

- Before sending frames, devices 'listen' to the collision domain until they detect that other devices are not sending.
- IF a collision occurs, 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.

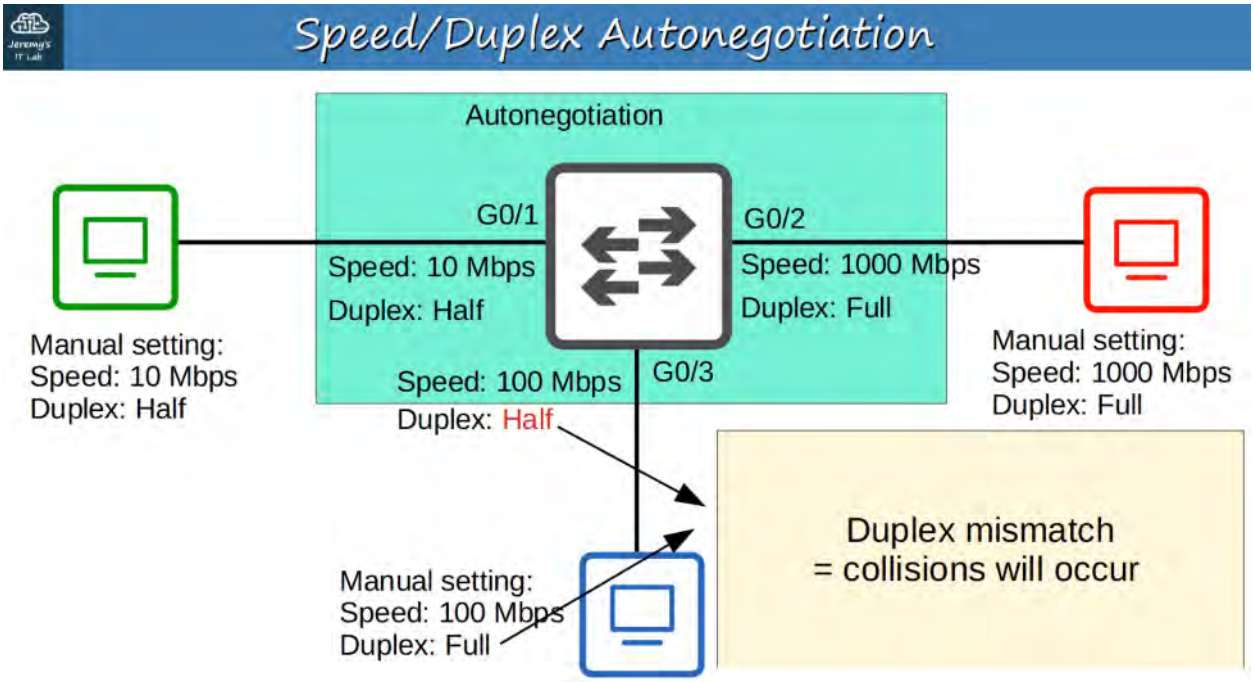
SWITCHES are more sophisticated than HUBS.
HUBS are Layer 1 Devices - Collisions are common and use CSMA/CD. SWITCHES are Layer 2 Devices - Collisions RARELY occur.



SPEED / DUPLEX AUTONEGOTIATION

- Interfaces that can run at different speeds (10/100 or 10/100/1000) have a default setting of SPEED AUTO and DUPLEX AUTO.
- Interfaces 'advertise' their capabilities to the neighbouring device, and they negotiate the best SPEED and DUPLEX settings they are both capable of.

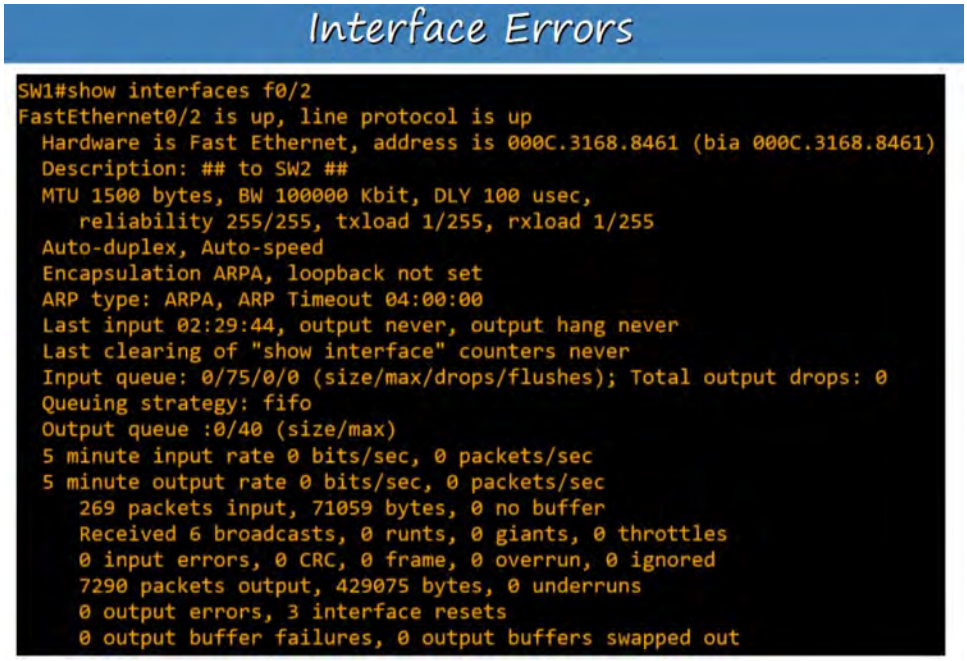
WHAT if AUTONEGOTIATION is DISABLED on the device connected to the SWITCH ?



- SPEED: The SWITCH will try to send at the speed that the other device is operating at. If it fails to send 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 great, it will use FULL DUPLEX.

INTERFACE COUNTERS AND ERRORS
Show using the:
// Privileged EXEC mode

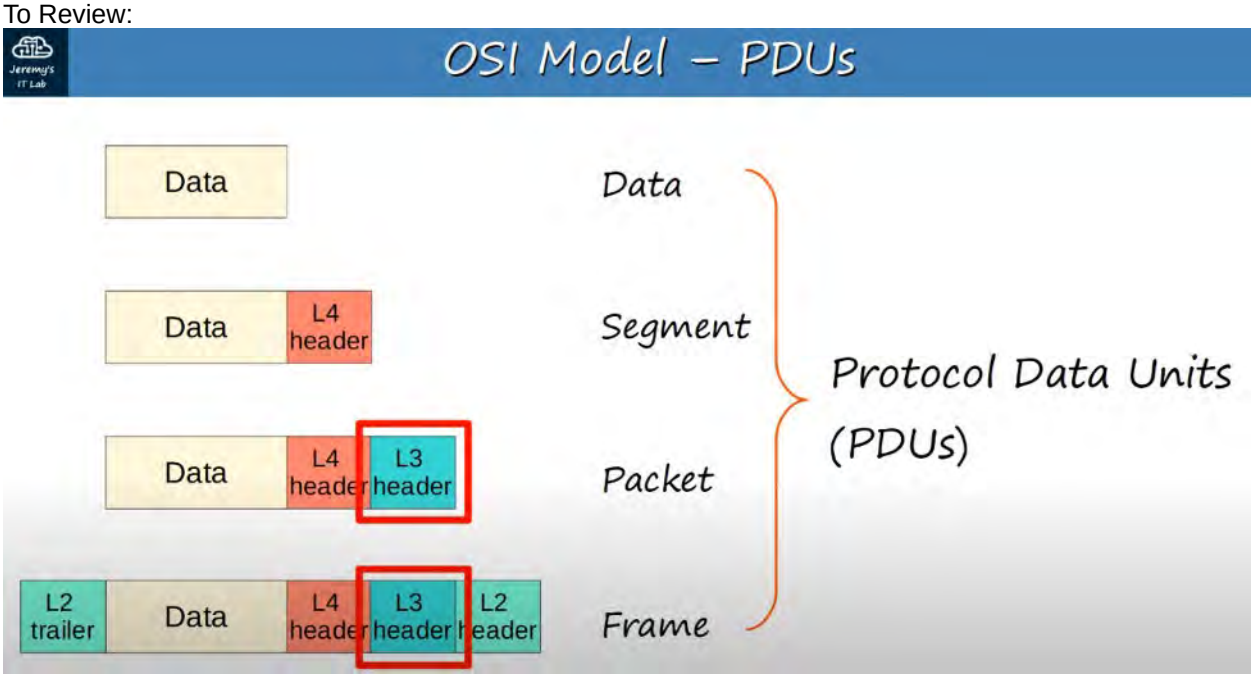
SW1#show interfaces
Error stats will be at the bottom.



Packets Received / Total bytes received.
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

10. THE IPv4 HEADER

INTERNET PROTOCOL version 4 HEADER or IPv4 HEADER
HEADER is used at LAYER 3 to help send data between devices on separate networks, even on other sides of the world over the Internet.
This is known as ROUTING.
THE IPv4 HEADER is used to ENCAPSULATE a TCP or UDP Segment.
To Review:



FIELDS OF THE IPv4 HEADER

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

FIELD	# OF BITS
VERSION	4
IHL	4
DSCP	6
ECN	2
TOTAL LENGTH	16
IDENTIFICATION	16

FIELD	# OF BITS
FLAGS	3
FRAGMENT OFFSET	13
TIME TO LIVE	8
PROTOCOL	8
HEADER CHECKSUM	16
SOURCE ADDRESS	32
DESTINATION ADDRESS	32
OPTIONS	320 Max

VERSION:

- LENGTH is 4 bits.
- IDs version of IP used (IPv4 or IPv6)
    - IPv4 = 0100 in Binary (Decimal 4)
    - IPv6 = 0110 in Binary (Decimal 6)

INTERNET HEADER LENGTH (IHL):

- LENGTH is 4 bits.
- Final field of IPv4 Header (Options) is variable in length so this field is necessary to indicate the total length of the header.
- IDs the length of the header in 4-BYTE INCREMENTS.
- The MINIMUM value is 5 (5 * 4-bytes = 20 bytes) - Empty OPTIONS Field
- The MAXIMUM value is 15 (15 * 4-bytes = 60 bytes)

MINIMUM IPv4 HEADER LENGTH = 20 Bytes! MAXIMUM IPv4 HEADER LENGTH = 60 Bytes!

DSCP (Differentiated Services Code Point):

- LENGTH is 6 bits.
- Used for QoS (Quality of Service)
- Used to prioritize delay-sensitive data (streaming voice, video, etc.)

ECN (Explicit Congestion Notification):

- LENGTH is 2 bits.
- Provides end-to-end (between two endpoints) notification of network congestion WITHOUT dropping packets.
- Optional feature that requires both endpoints, as well as the underlying network infrastructure to support it.

TOTAL LENGTH:

- LENGTH is 16 bits.
- Indicates the TOTAL length of the packet (L3 Header + L4 Segment)
- Measured in bytes (not 4-byte increments like IHL)
- Minimum value of 20 Bytes (IPv4 Header with NO encapsulated data)
- Maximum value of 65,535 (MAXIMUM 16-bit value) = 2^16

IDENTIFICATION:

- LENGTH is 16 bits.
- If a packet is fragmented due to being too large, this field is used to identify which packet the fragment belongs to.
- All fragments of the same packet will have their own IPv4 header with the same value in this field.

- Packets are fragmented, if larger than the MTU (Maximum Transmission Unit)
- The MTU is usually 1500 bytes (Max size of an Ethernet frame)
- Fragments are reassembled by the receiving host.

FLAGS:

- LENGTH is 3 bits
- Used to control/identify fragments.
- Bit 0: Reserved, always set to 0.
- Bit 1: Don't Fragment (DF bit), used to indicate a packet that should not be fragmented.
- Bit 2: More Fragments (MF bit), set to 1 if there are more fragments in the packet, set to 0 for the last fragment or NO fragments.

FRAGMENT OFFSET:

- LENGTH is 13 bits
- Used to indicated the position of the fragment within the original, unfragmented IP Packet.
- Allows fragmented packets to be reassembled even if the fragments arrive out of order.

TIME TO LIVE (TTL):

- LENGTH is 8 bits
- A router will drop a packet with a TTL of 0
- Used to prevent infinite loops
- Originally designed to indicated a packets maximum lifetime in seconds.
- In practice, indicates a 'hop count': each time the packet arrives at a router, the router decreases the TTL by 1.
- Recommended default TTL is 64.

PROTOCOL:

- LENGTH is 8 bits
- Indicates the protocol of the encapsulated Layer 4 PDU
- Value of 1 : ICMP
- Value of 6 : TCP
- Value of 17 : UDP
- Value of 89 : OSPF (Dynamic Routing Protocol)
- List of protocol numbers on Wikipedia : List of IP Protocol Numbers

HEADER CHECKSUM:

- LENGTH is 16 bits
- A calculated checksum used to check for errors in the IPv4 header.
- When a router receives a packet, it calculates the checksum of the header and compares it to the one in this field of a header.
- If they do not match, the router drops the packet.
- Used to check for ERRORS only in the IPv4 Header.
- IP relies on the encapsulated protocol to detect errors in the encapsulated data.
- Both TCP and UDP have their own checksum fields to detect errors in the encapsulated data.

SOURCE and DESTINATION:

- LENGTH is 32 bits each
- SOURCE IP = IPv4 ADDRESS of the Sender of the Packet.
- DESTINATION IP = IPv4 ADDRESS of the intended Receiver of the Packet.

OPTIONS:

- LENGTH is 0-320 bits
- Optional / Rarely Used
- If the IHL field is greater than 5, it means that Options are present.

11a. ROUTING FUNDAMENTALS : PART 1

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.

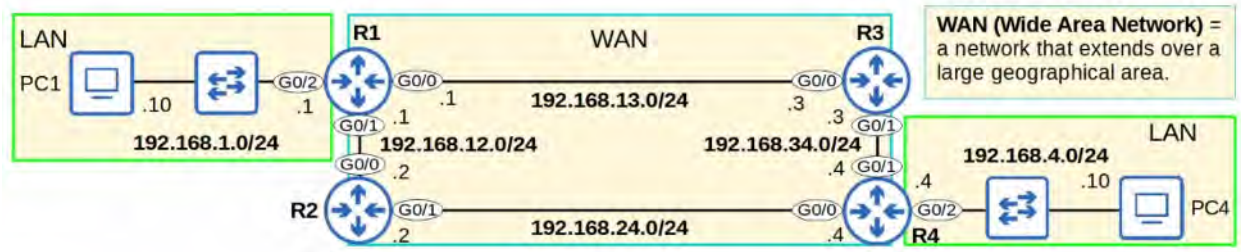
- ROUTERS store routes to all their known destinations in a ROUTING TABLE
- When ROUTERS receive PACKETS, they look in the ROUTING TABLE to find the best route to forward that packet.

There are two main routing methods (methods that routers use to learn routes):

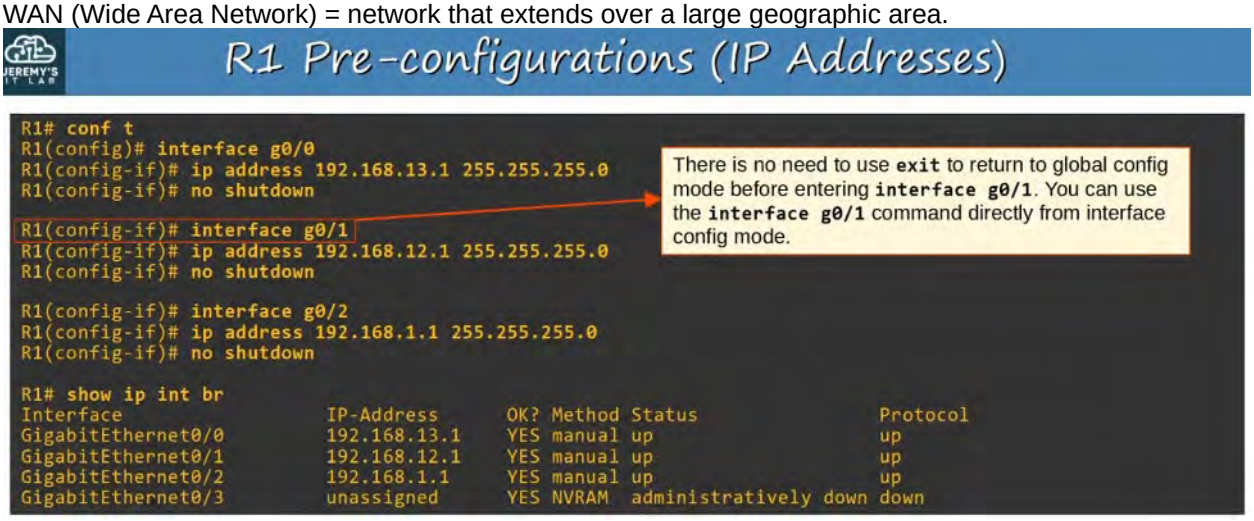
- DYNAMIC ROUTING : ROUTERS use Dynamic Routing Protocols (ie: OSPF) to share routing information with each other automatically and build their routing tables.
- STATIC ROUTING : A network engineer / Admin manually configures routes on the router.

A ROUTE tells the ROUTER :

- to send a packet to Destination X, you should send the pack 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) = network that extends over a large geographic area.



Routing Table (show ip route)

```
R1# 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
       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
```

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

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

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
```

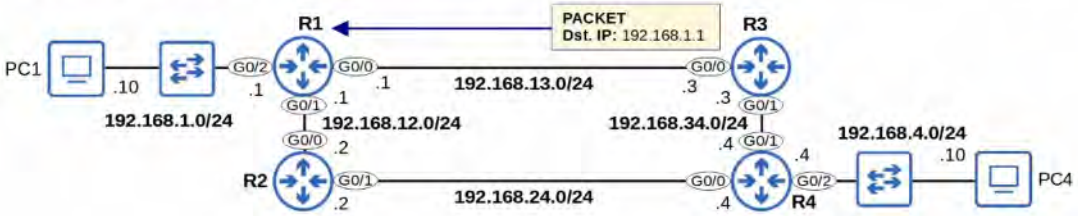
- A packet destined for **192.168.1.1** is matched by both routes:
  192.168.1.0/24
  192.168.1.1/32

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**.



Summary

- Routers store information about destinations they know in their **routing table**.
  →When they receive packets, the look in the routing table to find the best route to forward the packet.

- Each **route** in the routing table is an instruction:
  →To reach destinations in network X, send the packet to **next-hop** Y (the next router in the path to the destination).
  →If the destination is directly connected (**Connected** route) send the packet directly to the destination.
  →If the destination if your own IP address (**Local** route), receive the packet for yourself.
  *We will look at how **next-hops** work in the next video on **static routes**.

- When you configure an IP address on an interface and enable the interface, two routes are automatically added to the routing table:
  Connected route (code **C** in the routing table): A route to the network connected to the interface.
  →ie. if the interface's IP is **192.168.1.1/24**, the route will be to **192.168.1.0/24**.
  →Tells the router: "To send a packet to a destination in this network, send it out of the interface specified in the route".

  Local route (code **L** in the routing table): A route to the exact IP address configured on the interface.
  →ie. if the interface's IP is **192.168.1.1/24**, the route will be to **192.168.1.1/32**.
  →Tells the router: "Packets to this destination are for you. You should receive them for yourself (not forward them)".

- A route **matches** a destination if the packet's destination IP address is part of the network specified in the route.
  →ie. a packet to **192.168.1.60** is matched by a route to **192.168.1.0/24**, but not by a route to **192.168.0.0/24**.

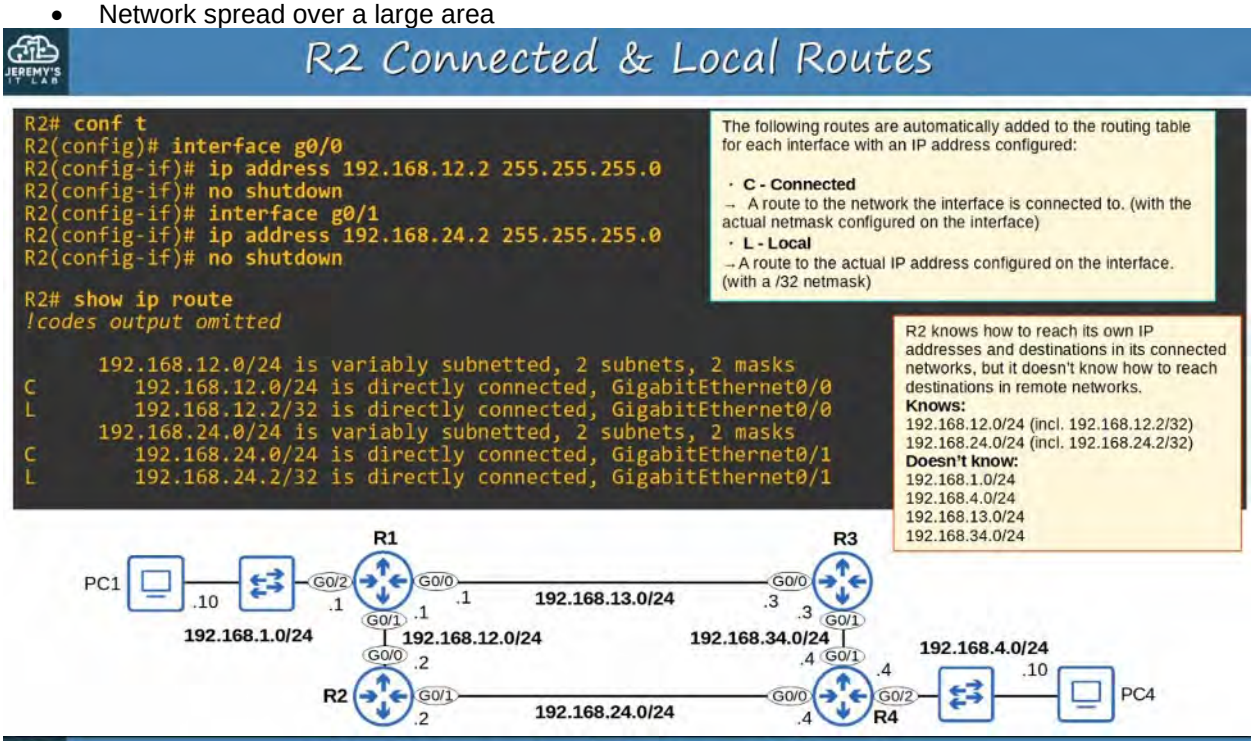
- If a router receives a packet and it doesn't have a route that matches the packet's destination, it will **drop** the packet.
  →This is different than switches, which **flood** frames if they don't have a MAC table entry for the destination.

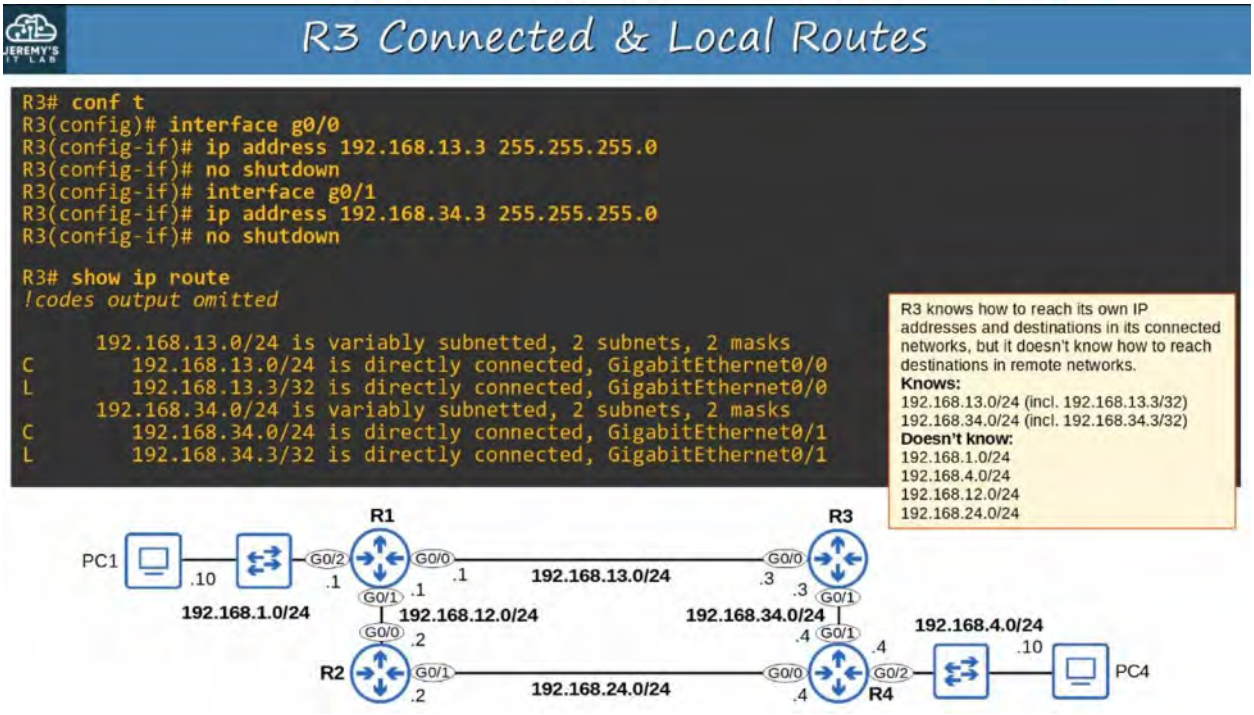
- If a router receives a packet and it has multiple routes that match the packet's destination, it will use the ***most specific*** *matching route* to forward the packet.
  →**Most specific** matching route = the matching route with the longest prefix length.
  →This is different than switches, which look for an **exact** match in the MAC address table to forward frames.

11b. STATIC ROUTING : PART 2

REVIEW:
SWITCHES forward traffic WITHIN LAN's ROUTERS forward traffic BETWEEN LAN's
WAN (Wide Area Network)

- Network spread over a large area





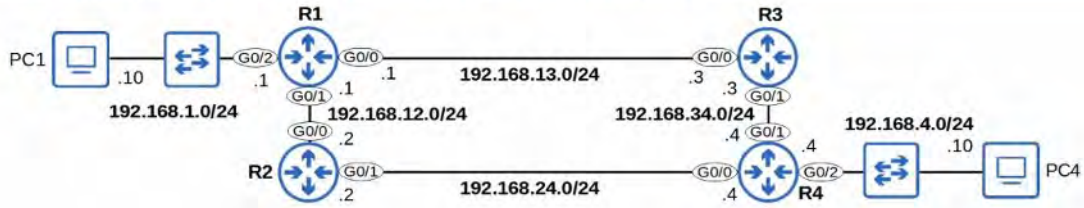
R4 Connected & Local Routes

```
R4(config)# interface g0/0
R4(config-if)# ip address 192.168.24.4 255.255.255.0
R4(config-if)# no shutdown
R4(config-if)# interface g0/1
R4(config-if)# ip address 192.168.34.4 255.255.255.0
R4(config-if)# no shutdown
R4(config-if)# interface g0/2
R4(config-if)# ip address 192.168.4.4 255.255.255.0
R4(config-if)# no shutdown

R4# show ip route
!codes output omitted
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, GigabitEthernet0/2
L        192.168.4.4/32 is directly connected, GigabitEthernet0/2
      192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.24.0/24 is directly connected, GigabitEthernet0/0
L        192.168.24.4/32 is directly connected, GigabitEthernet0/0
      192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.34.0/24 is directly connected, GigabitEthernet0/1
L        192.168.34.4/32 is directly connected, GigabitEthernet0/1
```

R4 knows how to reach its own IP addresses and destinations in its connected networks, but it doesn't know how to reach destinations in remote networks.

Knows:
192.168.4.0/24 (incl. 192.168.4.4/32)
192.168.24.0/24 (incl. 192.168.24.4/32)
192.168.34.0/24 (incl. 192.168.34.4/32)

Doesn't know:
192.168.1.0/24
192.168.12.0/24
192.168.13.0/24



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
```

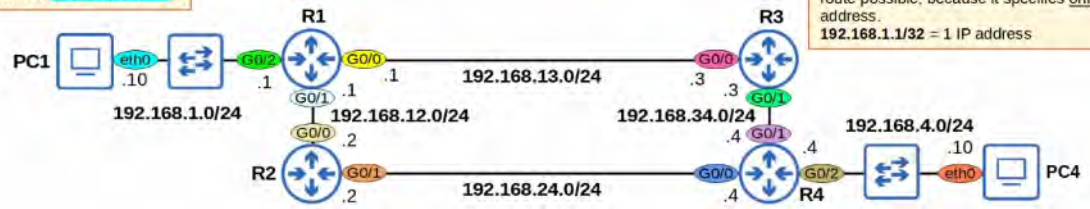
- The **default gateway** configuration is also called a **default route**.
  → It is a route to 0.0.0.0/0 = all netmask bits set to 0. Includes all addresses from 0.0.0.0 to 255.255.255.255.

- 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



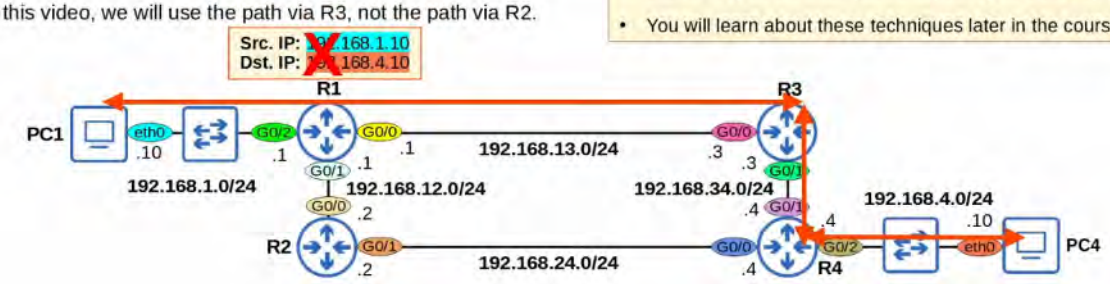
STATIC ROUTES:

Routing Packets: Static Routes

- When R1 receives the frame from PC1, it will de-encapsulate it (remove L2 header/trailer) and look at the inside packet.
- It will check the routing table for the most-specific matching route:
- R1 has no matching routes in its routing table.
  → It will drop the packet.
- To properly forward the packet, R1 needs a route to the destination network (192.168.4.0/24).
  → Routes are instructions: *To send a packet to destinations in network 192.168.4.0/24, forward the packet to next hop Y.*
- There are two possible path packets from PC1 to PC4 can take:
  1) PC1 → R1 → R3 → R4 → PC4
  2) PC1 → R1 → R2 → R4 → PC4
- In this video, we will use the path via R3, not the path via R2.

```
      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
```

- It is possible to configure the routers to:
  → *load-balance* between path **1)** and **2)**
  → Use path **1)** as the main path and path **2)** as a backup path
- You will learn about these techniques later in the course.



STATIC ROUTE CONFIGURATION:

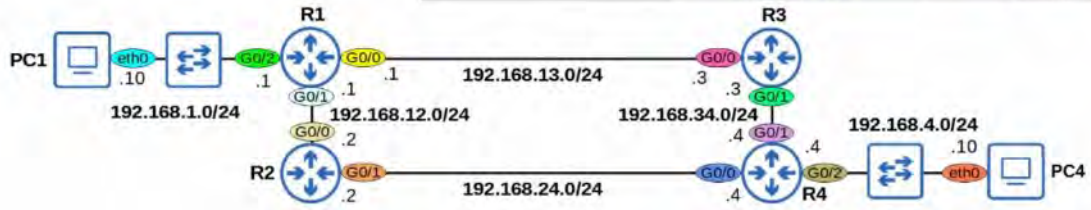
Static Route Configuration

- 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.

- To allow PC1 and PC4 to communicate with each other over the network, let's configure these **Static routes** on R1, R3, and R4.

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



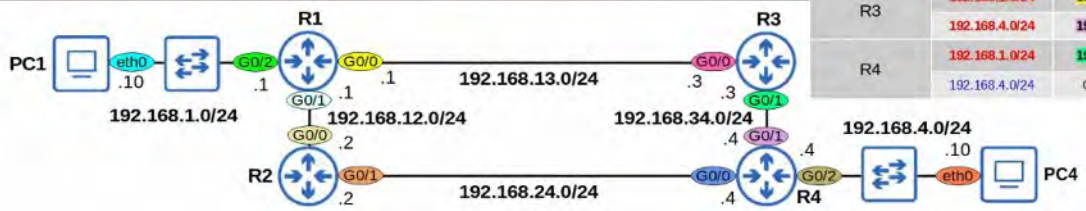
Static Route Configuration (R1)

```
R1(config)# ip route 192.168.4.0 255.255.255.0 192.168.13.3

R1(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

      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
S     192.168.4.0/24 [1/0] via 192.168.13.3
      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
```

`R1(config)# ip route ip-address netmask next-hop`

The [1/0] displayed in static routes means:
[Administrative Distance/Metric]
We will cover these concepts later in the course.

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

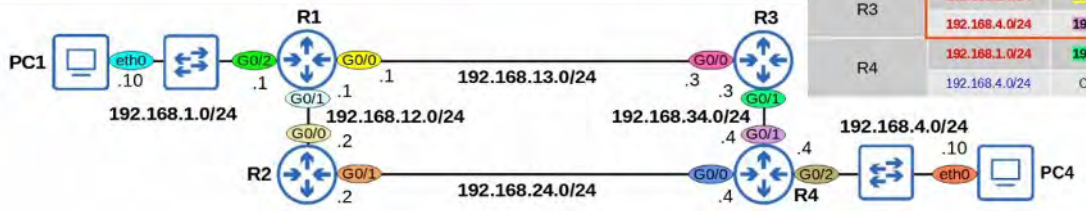


Static Route Configuration (R3)

```
R3(config)# ip route 192.168.1.0 255.255.255.0 192.168.13.1
R3(config)# ip route 192.168.4.0 255.255.255.0 192.168.34.4

R3(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 [1/0] via 192.168.13.1
S     192.168.4.0/24 [1/0] via 192.168.34.4
      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.13.0/24 is directly connected, GigabitEthernet0/0
L        192.168.13.3/32 is directly connected, GigabitEthernet0/0
      192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.34.0/24 is directly connected, GigabitEthernet0/1
L        192.168.34.3/32 is directly connected, GigabitEthernet0/1
```

`R3(config)# ip route ip-address netmask next-hop`

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

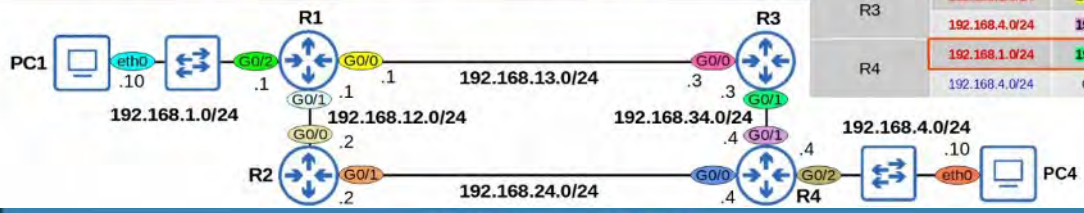


Static Route Configuration (R4)

```
R4(config)# ip route 192.168.1.0 255.255.255.0 192.168.34.3

R4(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 [1/0] via 192.168.34.3
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, GigabitEthernet0/2
L        192.168.4.4/32 is directly connected, GigabitEthernet0/2
      192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.24.0/24 is directly connected, GigabitEthernet0/0
L        192.168.24.4/32 is directly connected, GigabitEthernet0/0
      192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.34.0/24 is directly connected, GigabitEthernet0/1
L        192.168.34.4/32 is directly connected, GigabitEthernet0/1
```

`R4(config)# ip route ip-address netmask next-hop`

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



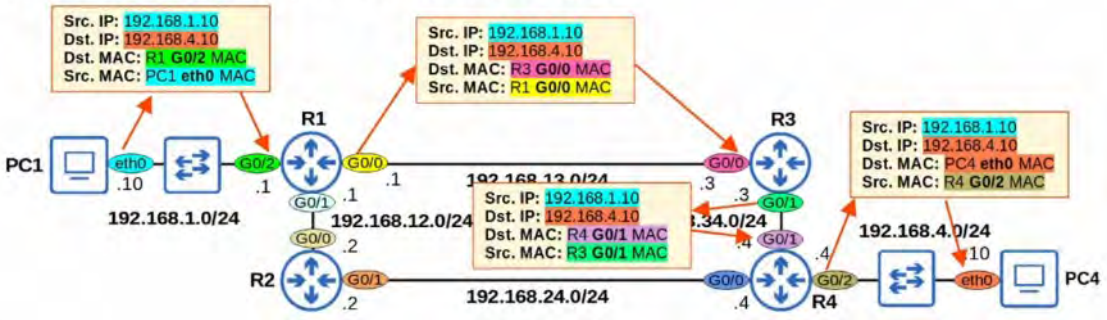
PC1 ↔ PC4

```
PC1:~$ ping 192.168.4.10
PING 192.168.4.10 (192.168.4.10): 56 data bytes
64 bytes from 192.168.4.10: seq=0 ttl=42 time=8.745 ms
64 bytes from 192.168.4.10: seq=1 ttl=42 time=4.423 ms
64 bytes from 192.168.4.10: seq=2 ttl=42 time=3.428 ms
64 bytes from 192.168.4.10: seq=3 ttl=42 time=3.544 ms
64 bytes from 192.168.4.10: seq=4 ttl=42 time=3.520 ms
^C
--- 192.168.4.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.428/4.732/8.745 ms
```

If the ping is successful, that means there is two-way reachability.
PC1 can reach PC4, and PC4 can reach PC1.

Packet traveling from PC1 to PC4:

*we will examine this step-by-step in the "Life of a Packet" video



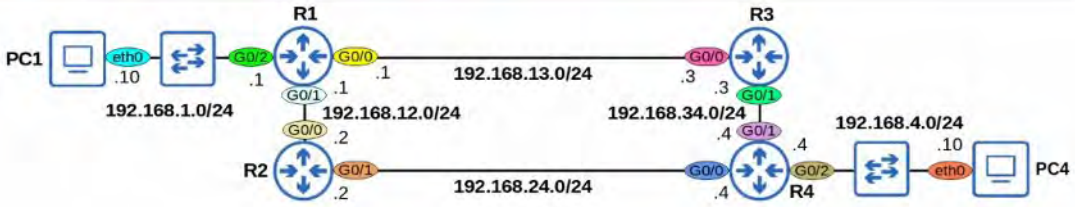
STATIC ROUTE CONFIGURATION with *exit-interface*

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
```

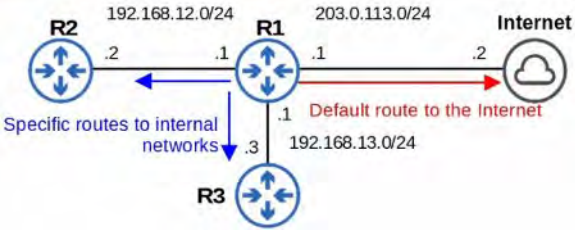
- Static routes in which you specify only the *exit-interface* rely on a feature called **Proxy ARP** to function.
- This is usually not a problem, but generally you can stick to *next-hop* or *exit-interface next-hop*.
- Neither is 'better' than the other: use which you prefer.



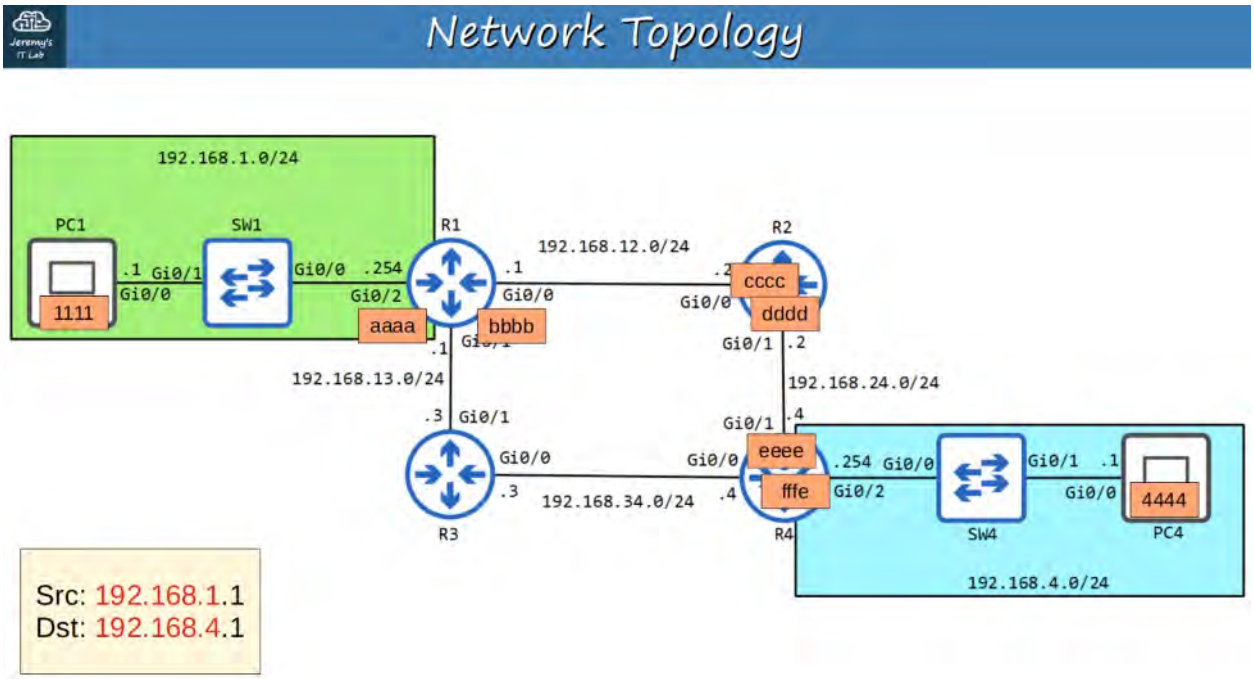
DEFAULT ROUTE

Default Route

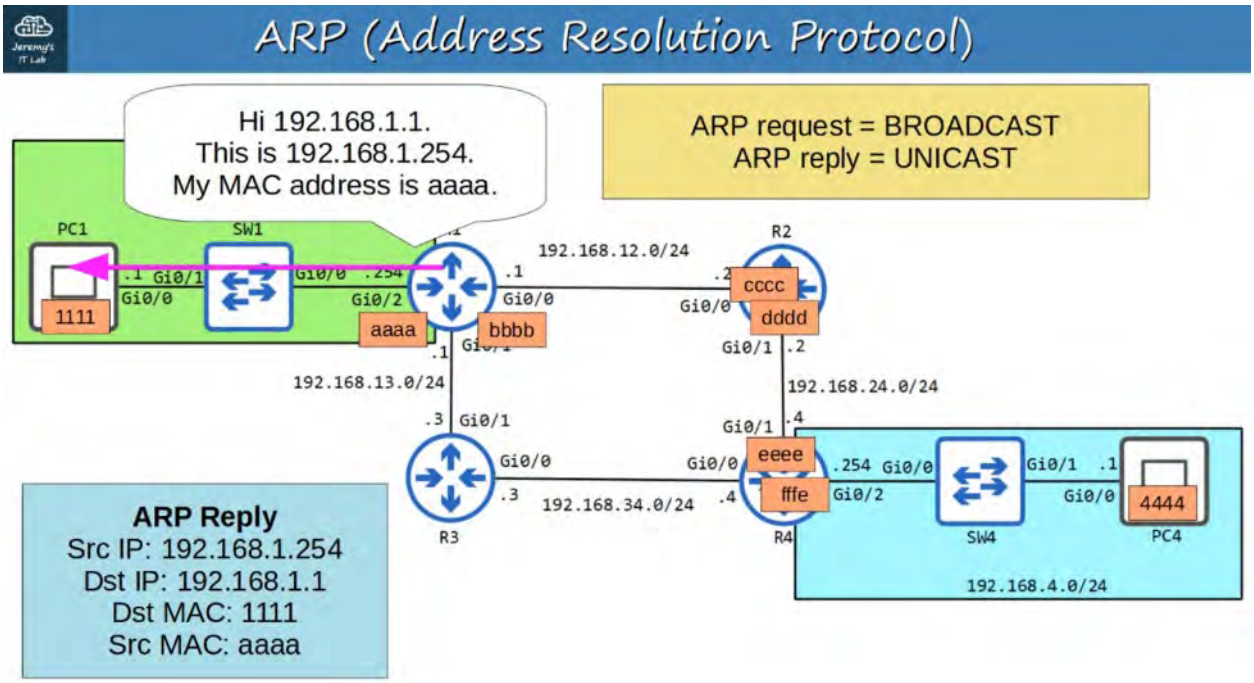
```
R1(config)# ip route 0.0.0.0 0.0.0.0 203.0.113.2
R1(config)# do show ip route
!most codes omitted
        ia - IS-IS inter area, * - candidate default, U - per-user static route
!most codes omitted
Gateway of last resort is 203.0.113.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 203.0.113.2
S     10.0.0.0/8 [1/0] via 192.168.12.2
S     172.16.0.0/16 [1/0] via 192.168.13.3
      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
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C        203.0.113.0/24 is directly connected, GigabitEthernet0/2
L        203.0.113.1/32 is directly connected, GigabitEthernet0/2
```



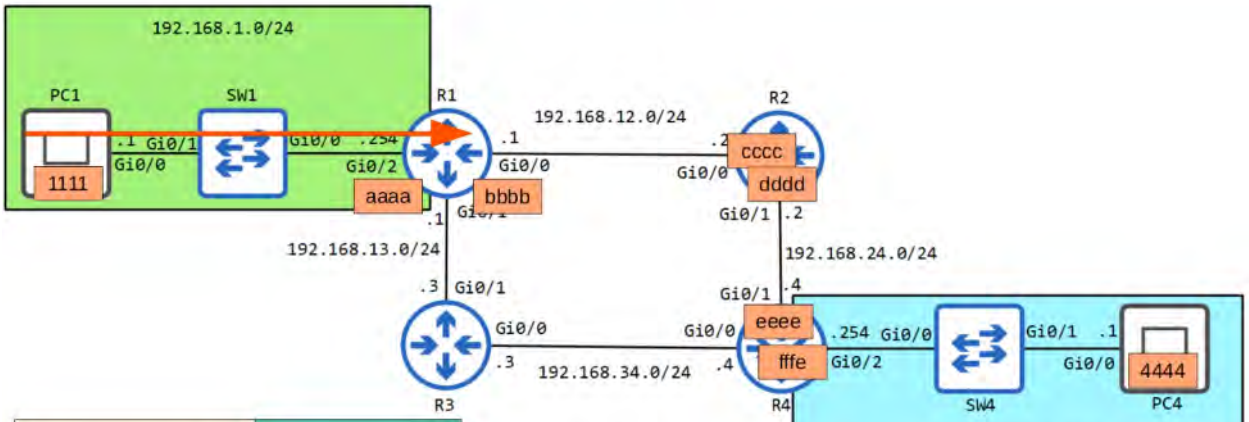
12. LIFE OF A PACKET



EACH Network device's interfaces have a UNIQUE MAC Addresses.
In a TCP HEADER
SOURCE IP ADDRESS comes before DESTINATION IP ADDRESS
while...
in an ETHERNET HEADER
DESTINATION MAC ADDRESS comes before SOURCE MAC ADDRESS

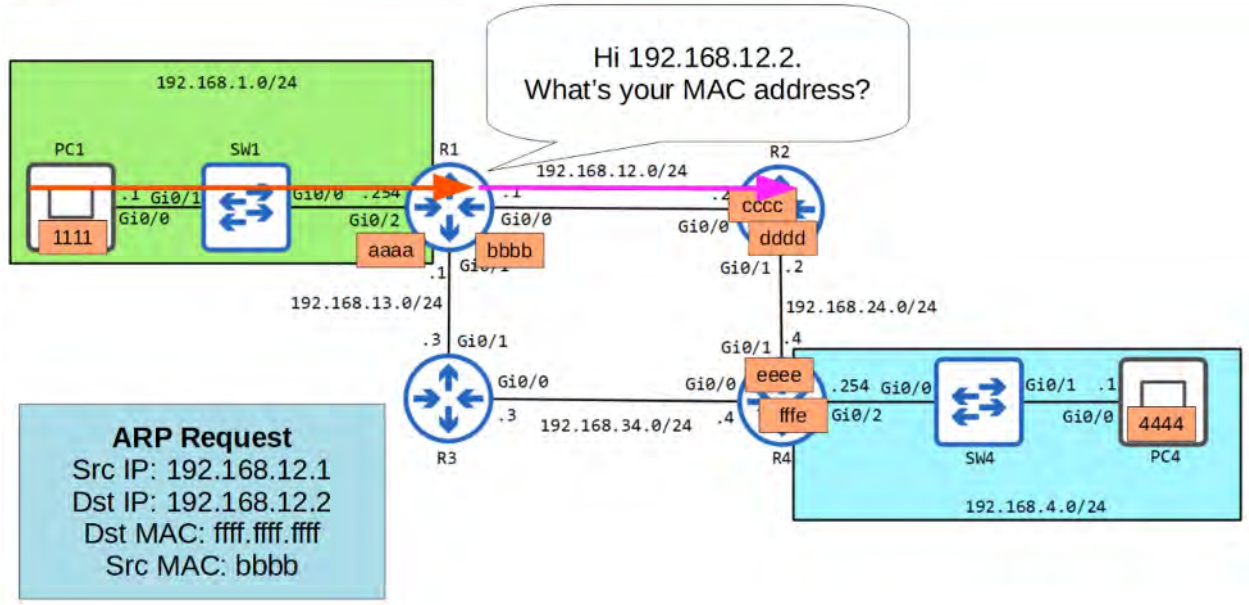


PC1 → R1

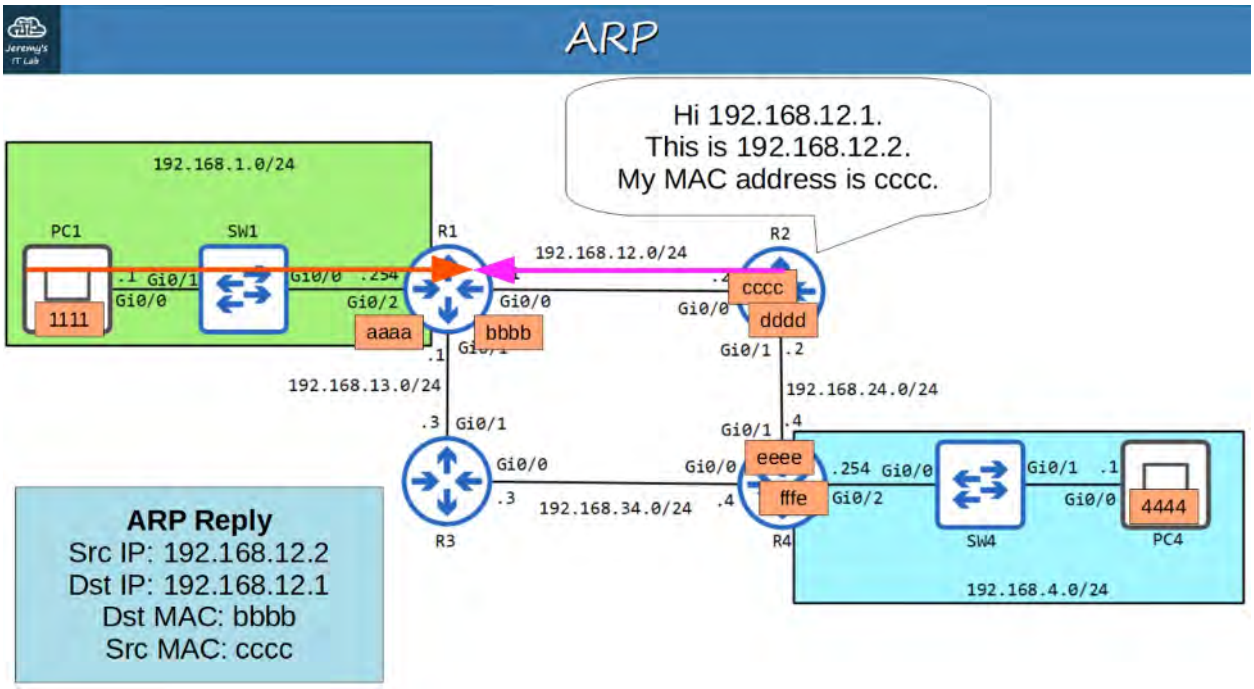


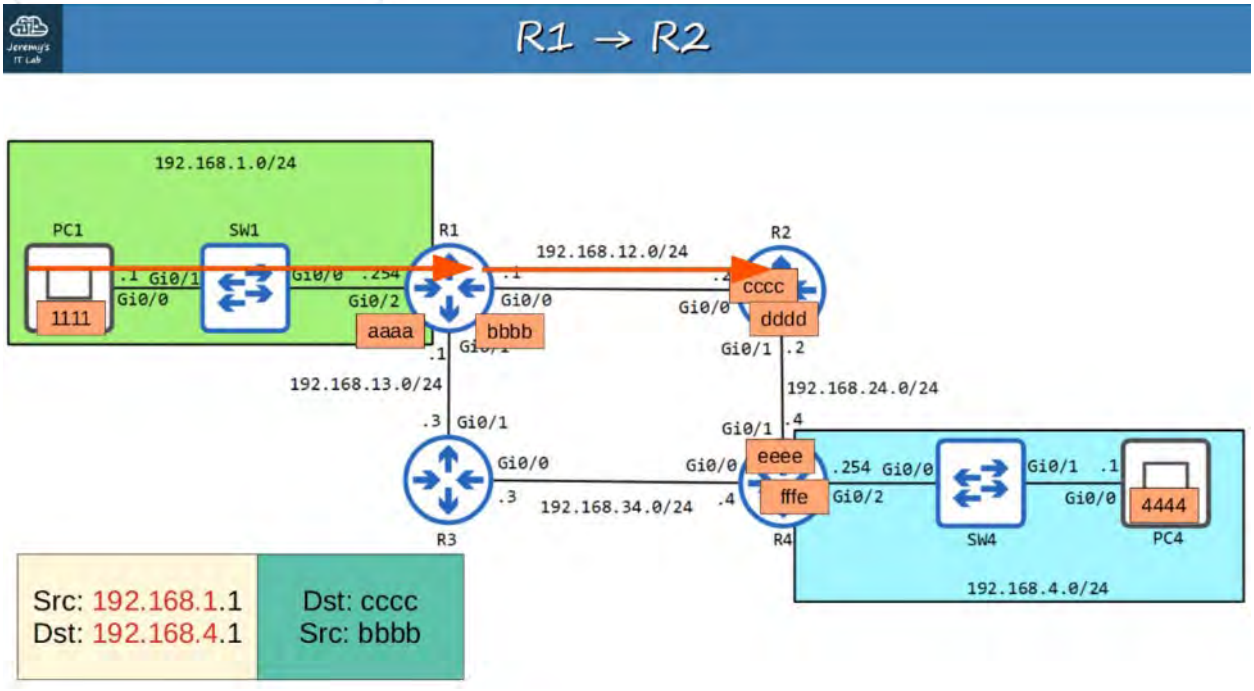
Src: 192.168.1.1 Dst: 192.168.4.1	Dst: aaaa Src: 1111

ARP

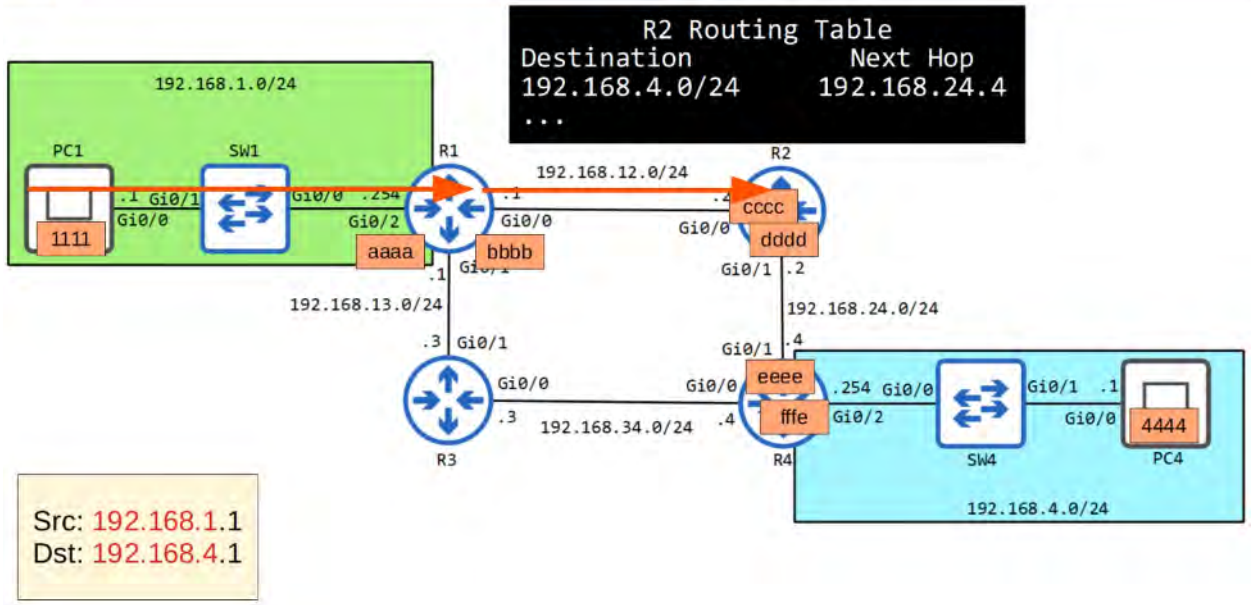


ARP Request
Src IP: 192.168.12.1
Dst IP: 192.168.12.2
Dst MAC: ffff.ffff.ffff
Src MAC: bbbb

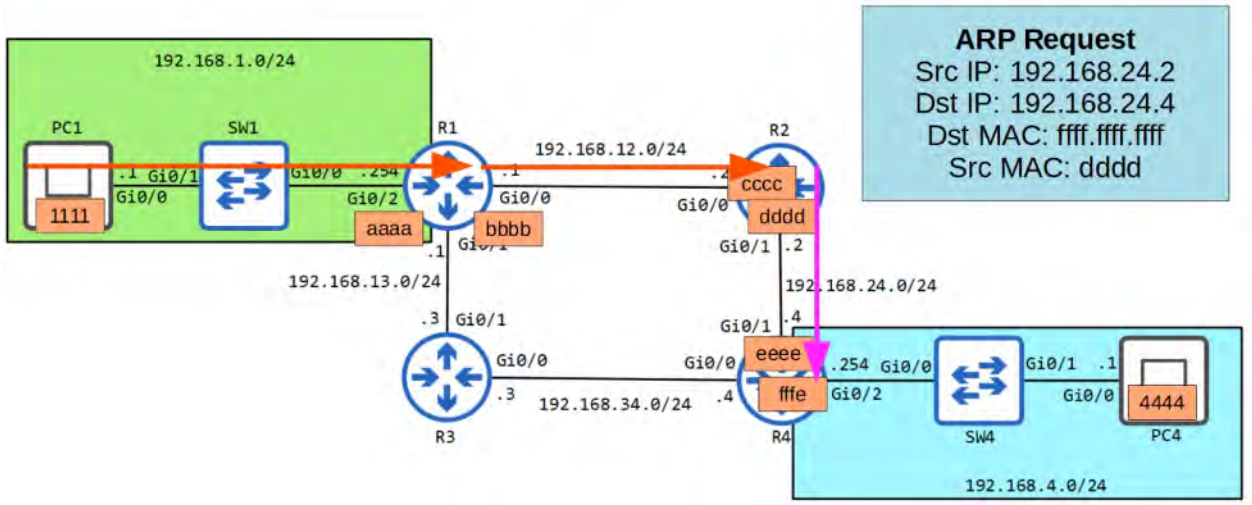

ARP
Hi 192.168.12.1.
This is 192.168.12.2.
My MAC address is cccc.
192.168.1.0/24
PC1
SW1
R1
R2
1111
aaaa
bbbb
cccc
dddd
192.168.12.0/24
192.168.13.0/24
192.168.24.0/24
192.168.34.0/24
R3
R4
eeee
fffe
SW4
4444
PC4
192.168.4.0/24
ARP Reply
Src IP: 192.168.12.2
Dst IP: 192.168.12.1
Dst MAC: bbbb
Src MAC: cccc


R1 → R2
192.168.1.0/24
PC1
SW1
R1
R2
1111
aaaa
bbbb
cccc
dddd
192.168.12.0/24
192.168.13.0/24
192.168.24.0/24
192.168.34.0/24
R3
R4
eeee
fffe
SW4
4444
PC4
192.168.4.0/24
Src: 192.168.1.1
Dst: 192.168.4.1
Dst: cccc
Src: bbbb

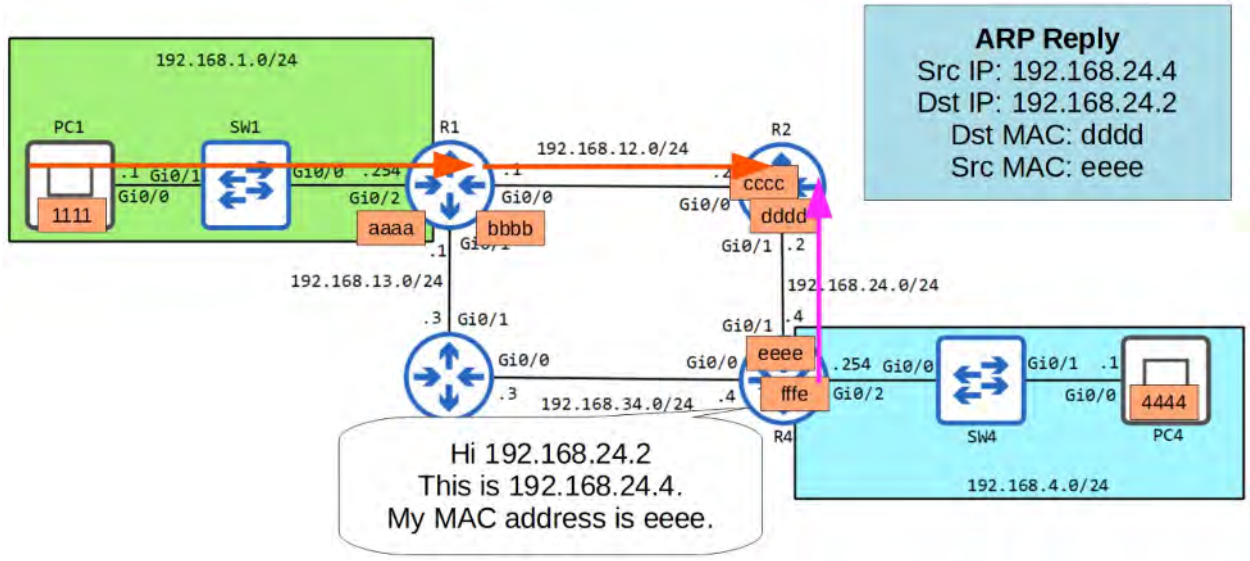
R2 → R4


R2 Routing Table
Destination Next Hop
192.168.4.0/24 192.168.24.4
...
192.168.1.0/24
PC1
SW1
R1
R2
1111
aaaa
bbbb
cccc
dddd
192.168.12.0/24
192.168.13.0/24
192.168.24.0/24
192.168.34.0/24
R3
eeee
fffe
R4
SW4
PC4
4444
192.168.4.0/24
Src: 192.168.1.1
Dst: 192.168.4.1

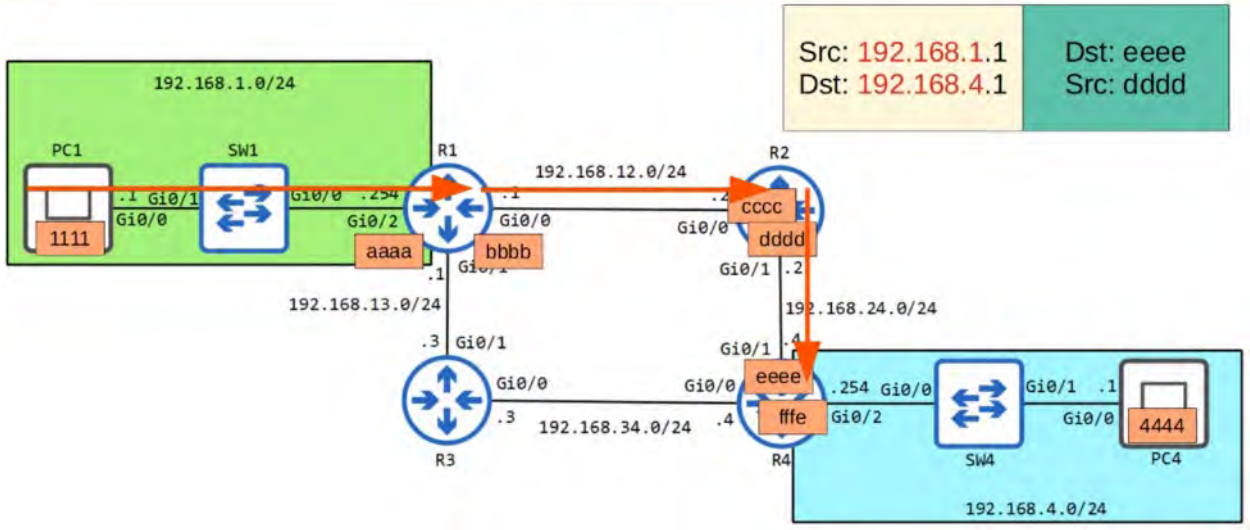
ARP


ARP Request
Src IP: 192.168.24.2
Dst IP: 192.168.24.4
Dst MAC: ffff.ffff.ffff
Src MAC: dddd
192.168.1.0/24
PC1
SW1
R1
R2
1111
aaaa
bbbb
cccc
dddd
192.168.12.0/24
192.168.13.0/24
192.168.24.0/24
192.168.34.0/24
R3
eeee
fffe
R4
SW4
PC4
4444
192.168.4.0/24

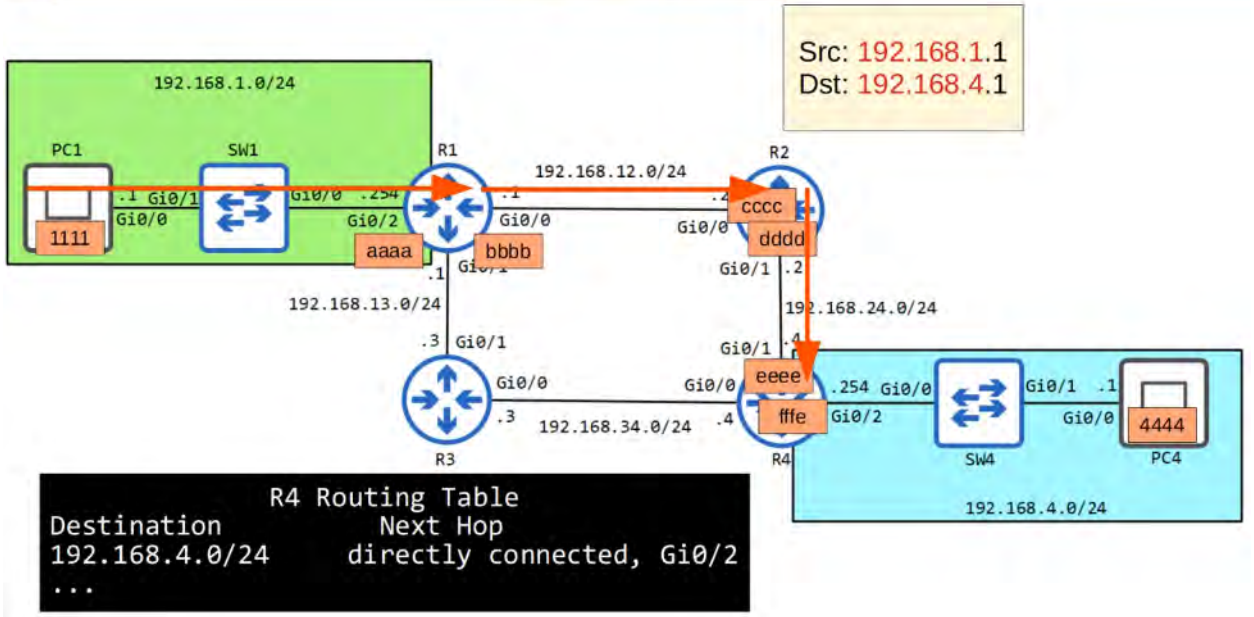
ARP


ARP Reply
Src IP: 192.168.24.4
Dst IP: 192.168.24.2
Dst MAC: dddd
Src MAC: eeee
192.168.1.0/24
PC1
SW1
R1
R2
1111
aaaa
bbbb
cccc
dddd
eeee
fffe
192.168.12.0/24
192.168.13.0/24
192.168.24.0/24
192.168.34.0/24
192.168.4.0/24
SW4
PC4
4444
R4
Hi 192.168.24.2
This is 192.168.24.4.
My MAC address is eeee.

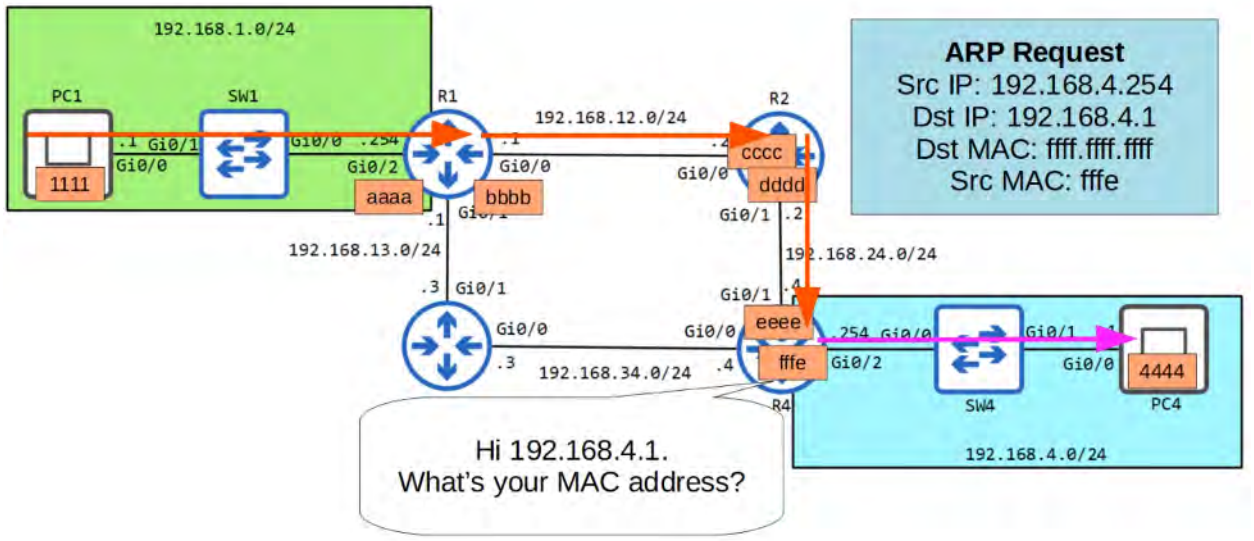
R2 → R4


Src: 192.168.1.1
Dst: 192.168.4.1
Dst: eeee
Src: dddd
192.168.1.0/24
PC1
SW1
R1
R2
R3
R4
1111
aaaa
bbbb
cccc
dddd
eeee
fffe
192.168.12.0/24
192.168.13.0/24
192.168.24.0/24
192.168.34.0/24
192.168.4.0/24
SW4
PC4
4444

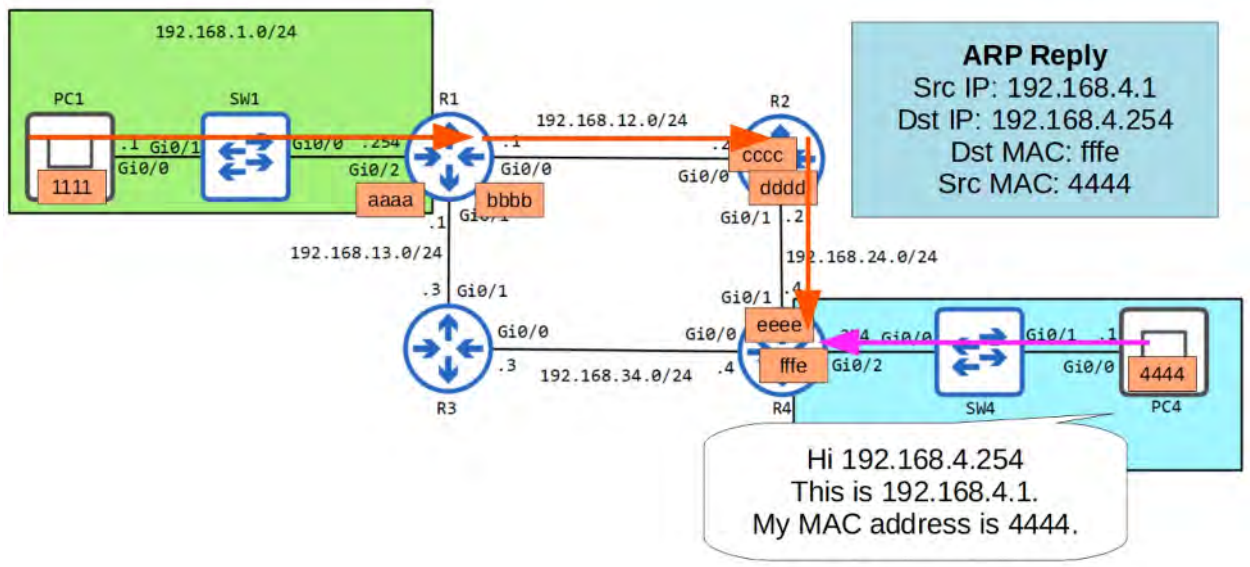
R2 → R4


Src: 192.168.1.1
Dst: 192.168.4.1
192.168.1.0/24
PC1
SW1
R1
R2
192.168.12.0/24
1111
aaaa
bbbb
cccc
dddd
192.168.13.0/24
192.168.24.0/24
R3
eeee
fffe
192.168.34.0/24
R4
SW4
PC4
4444
192.168.4.0/24
R4 Routing Table
Destination Next Hop
192.168.4.0/24 directly connected, Gi0/2
...

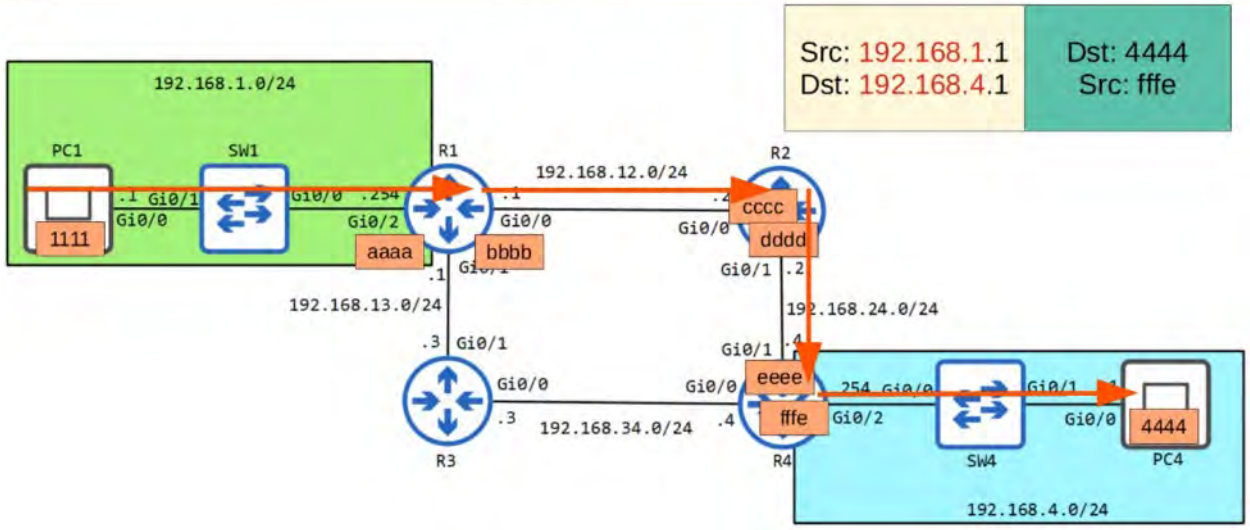
ARP

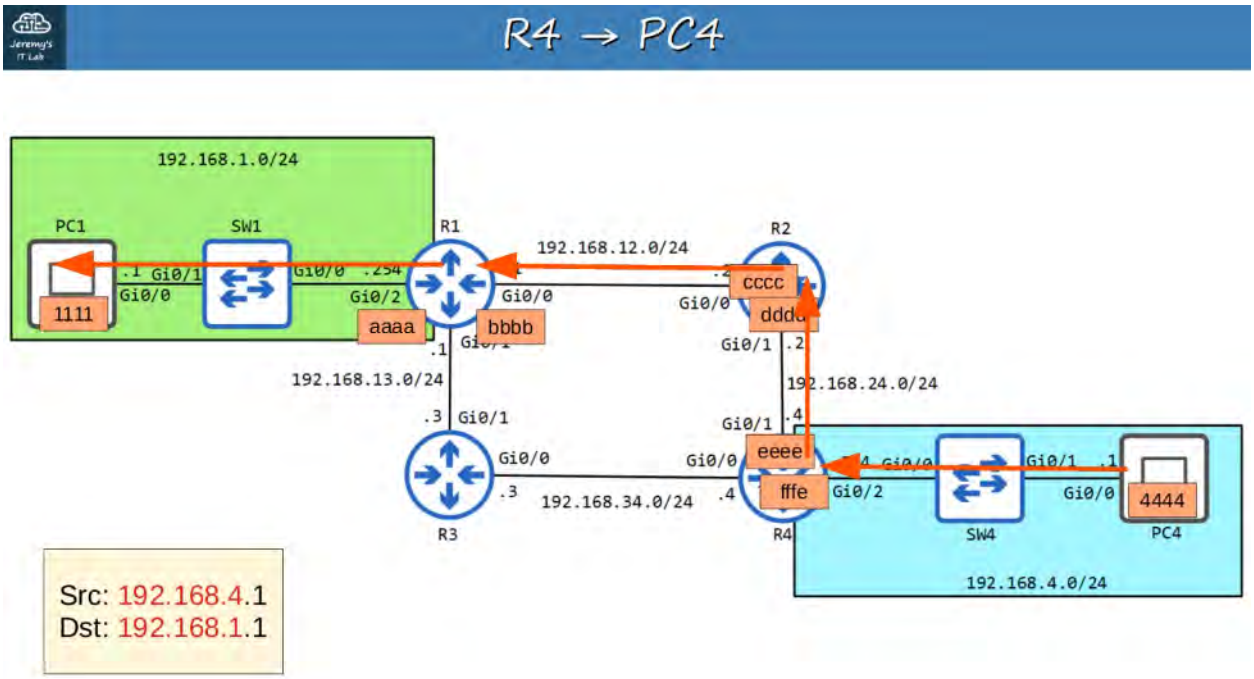

192.168.1.0/24
PC1
SW1
R1
R2
192.168.12.0/24
1111
aaaa
bbbb
cccc
dddd
192.168.13.0/24
192.168.24.0/24
R3
eeee
fffe
192.168.34.0/24
R4
SW4
PC4
4444
192.168.4.0/24
ARP Request
Src IP: 192.168.4.254
Dst IP: 192.168.4.1
Dst MAC: ffff.ffff.ffff
Src MAC: fffe
Hi 192.168.4.1.
What's your MAC address?

ARP


192.168.1.0/24
PC1
SW1
R1
R2
1111
aaaa
bbbb
cccc
dddd
eeee
ffff
R3
R4
SW4
PC4
4444
192.168.12.0/24
192.168.13.0/24
192.168.24.0/24
192.168.34.0/24
Gi0/0
Gi0/1
Gi0/2
.1
.2
.3
.4
.254
ARP Reply
Src IP: 192.168.4.1
Dst IP: 192.168.4.254
Dst MAC: fffe
Src MAC: 4444
Hi 192.168.4.254
This is 192.168.4.1.
My MAC address is 4444.

R4 → PC4


Src: 192.168.1.1
Dst: 192.168.4.1
Dst: 4444
Src: fffe
192.168.1.0/24
PC1
SW1
R1
R2
1111
aaaa
bbbb
cccc
dddd
eeee
ffff
R3
R4
SW4
PC4
4444
192.168.12.0/24
192.168.13.0/24
192.168.24.0/24
192.168.34.0/24
192.168.4.0/24
Gi0/0
Gi0/1
Gi0/2
.1
.2
.3
.4
.254



When a HOST sends a packet to another HOST, the SOURCE or DESTINATION IP doesn't change - even though ROUTERS may change the ETHERNET HEADER (SRC/DEST MAC ADDRESSES).

13. SUBNETTING : PART 1

IPv4 Address Classes

Class	First octet (binary)	First octet range (decimal)
A	0xxxxxxx	0 - 127
B	10xxxxxx	128 - 191
C	110xxxxx	192 - 223
D	1110xxxx	224 - 239
E	1111xxxx	240 - 255

HOWEVER, only Class A, B, C Addresses can be assigned to a device as an IP Address.
CLASS PREFIX LENGTH
A /8 B /16 C /24

IPv4 Address Classes

Class	Leading bits	Size of *network number* bit field	Size of *rest* bit field	Number of networks	Addresses per network
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})
Class C	110	24	8	2,097,152 (2^{21})	256 (2^8)

The IANA (Internet Assigned Numbers Authority) assigns IPv4 addresses/networks to companies based on their size.
The problem with 'CLASSFUL' assignment is that it led to IP Address wastefulness.
Example: A company requiring 5000 address was assigned a CLASS B IP, leaving 60000+ addresses unused.

The IETF (Internet Engineering Task Force) introduce CIDR in 1993 to replace the "classful" addressing system.
CIDR (Classless Inter-Domain Routing) removed the requirements of CLASS A, B, and C regarding size.

- This allowed larger networks to be split into smaller networks, allowing greater efficiency.
- These smaller networks are called "SUB-NETWORKS" or "SUBNETS"

HOW MANY USABLE ADDRESSES ARE THERE IN EACH NETWORK?
REMEMBER:
2^n - 2 = Usable Address n = number of host bits
CIDR PRACTICE!
203.0.113.0/25
/25 means the Subnetwork bit is 25 bits
203 . 0 . 113 . 0 is written in binary as :
1100 1011 . 0000 0000 . 0111 0001 . 0 | 000 0000
(Subnet prefix is the first 25 bits)

Flipping all the bits to 1's, we get the SUBNET MASK for /25:
1111 1111 . 1111 1111 . 1111 1111 . 1 | 000 0000
which is equal to:
255.255.255.128 (because the last octet is 1000 0000 = 128 in binary)
SO - the based on previous definition of USABLE ADDRESSES, the number of hosts for 203.0.113.0 /25 is:
2^(7 bits) or (128) - 2 = 126 hosts.

What about /28 ?
203 . 0 . 113 . 0 is written in binary as :
1100 1011 . 0000 0000 . 0111 0001 . 0000 | 0000
(Subnet prefix is the first 28 bits)
flipping all the bits to 1's, we get the SUBNET MASK for /28:
1111 1111 . 1111 1111 . 1111 1111 . 1111 | 0000
which is equal to:
255.255.255.240 (because the last octet is 1111 0000) = 128+64+32+16 = (128+32) + (64+16) = 160 + 80 = 240
The SUBNET MASK for /28 is 255.255.255.240 which has 16 hosts / group (2 * 4 bits = 16) - 2 Reserved IPs for Network and Broadcast

---

SUBNETTING CHEATSHEET:

Group Size	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Subnet Mask	128	192	224	240	248	252	254	255
CIDR	/25	/26	/27	/28	/29	/30	/31	/32
3rd Octet	/17	/18	/19	/20	/21	/22	/23	/24
2nd Octet	/9	/10	/11	/12	/13	/14	/15	/16
1st Octet	/1	/2	/3	/4	/5	/6	/7	/8

---

1. Use a given CIDR/Mask to find column on Cheat Sheet

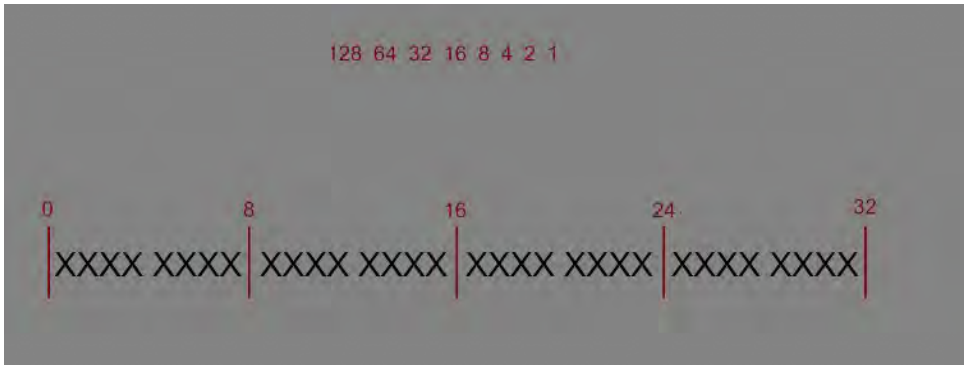
a) CIDR/Subnet Mask map to each other
b) Locate Group Size
c) Increase by Group Size until you PASS the Target IP (not less or equal !)
d) If passing the Target IP reaches 256, increase the Octet BEFORE it by one and current Octet becomes 0 : IF NECESSARY
Example: 10.2.2.256 → 10.2.3.0

2. Number BEFORE Target IP is NETWORK ID
3. Number AFTER Target IP is NEXT NETWORK
4. IP Address BEFORE Next Network is BROADCAST
5. IP Address AFTER Network ID is First Host
6. IP Address BEFORE Broadcast IP is Last Host
7. Group Size is total # of IP Addresses
    - Don't forget to subtract 2 for USABLE #

Solving CIDR/Subnet for 3rd Octet IPs :
Every number LEFT of 3rd Octet is 255. Every number RIGHT of 3rd Octet is 0
Example: 10.4.77.188 / 19 → Subnet : 255.255.224.0
You use the SAME process as above except when finding Target IPs, you use the 3rd Octet for your Target.
Example: 10.4.77.188 /19 → Subnet : 255.255.224.0
256 - 224 = 32 so…
Using 32, we step through the address blocks 0, 32,64, and 96. Since 77 is between 64 and 96, there's our range.

Network: 10.4.64.0 (Start / First Block)
Next: 10.4.96.0 (Second Block) …
Number of IP Addresses is : 2^(32-CIDR). In this example 2^13 = 8192
Solving for 2nd and 1st Octet is the same as above, keeping in mind the Octet column is USED to check for the Target number of a given address.

Alternative method to "Cheat Sheet"



1. Find the "magic octet" where a given IP /Prefix lies, from the bit chart shown (boundary digits are inclusive of the octet preceding them)
2. Count the number of network bits (left to right) in that octet and count the same amount, using the red bit slot chart. This'll be your address group size.
3. Subtract that number from 256 to find your Subnet Mask number used in the "magic octet" (any octet LEFT of that "magic octet" will be 255, everything RIGHT of that octet will be 0)
4. Divide whatever IP octet number is in the "magic octet" by the address group size.

- If there is a remainder, multiple the whole integer by the address group size - your Base Network Address is that value, with every octet to the right of that as all 0's
- If there is NO remainder, the IP number in the "magic octet" IS the Base Network Address is that value, with every octet to the right of that as all 0's

5. The Base Broadcast Number will be Network Base Number + Group Size - 1 in the "magic octet", every value to the right of that octet will be 255.
6. Number of subnets is (2 to the power of the number of network bits in the "magic octet". ** 2^8 or 256 is equal to 0 **)
7. Total Useable Hosts size is (2 to the power of (32 - Prefix Length) -2)

Example 1:
154 . 219 . 154 . 180 /20

Third Octet = Magic

Address Group Size = 16 (L/R count of 4)
256 - 16 = 240 therefore Subnet Mask is 255.255.240.0

Divide 3rd digit / Address Group Size (16)
154 / 16 = 9 (with remainder)
9 * 16 = 144 (Base Network #)

Network : 154 . 219 . 144 . 0

Broadcast Base # = 144 + 16 - 1 = 159

Broadcast : 154. 219 . 159 . 255

Subnets = 2^4 network bits = 16
Total Host Size = (2^(32 - 20))-2 = 4094

Example 2:
84 . 75 . 21 .6 /10

Second Octet = Magic

Address Group Size = 64
256 - 64 = 192

Subnet = 255.192.0.0

75 / 64 = 1 + remainder
1 * 64 = 64 (Base Network #)

Network : 84.64.0.0

Broadcast Base # = 64 + 64 -1 = 127

Broacast : 84.127.255.255

Subnets : 2^2 = 4 Subnets
Total Host Size = (2^(32-10))-2 = 4194302

14. SUBNETTING : PART 2

CLASS C NETWORKS

Subnets/Hosts (Class C)

Prefix Length	Number of Subnets	Number of Hosts
/25	2	126
/26	4	62
/27	8	30
/28	16	14
/29	32	6
/30	64	2
/31	128	0 (2)
/32	256	0 (1)

CLASS B NETWORKS

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

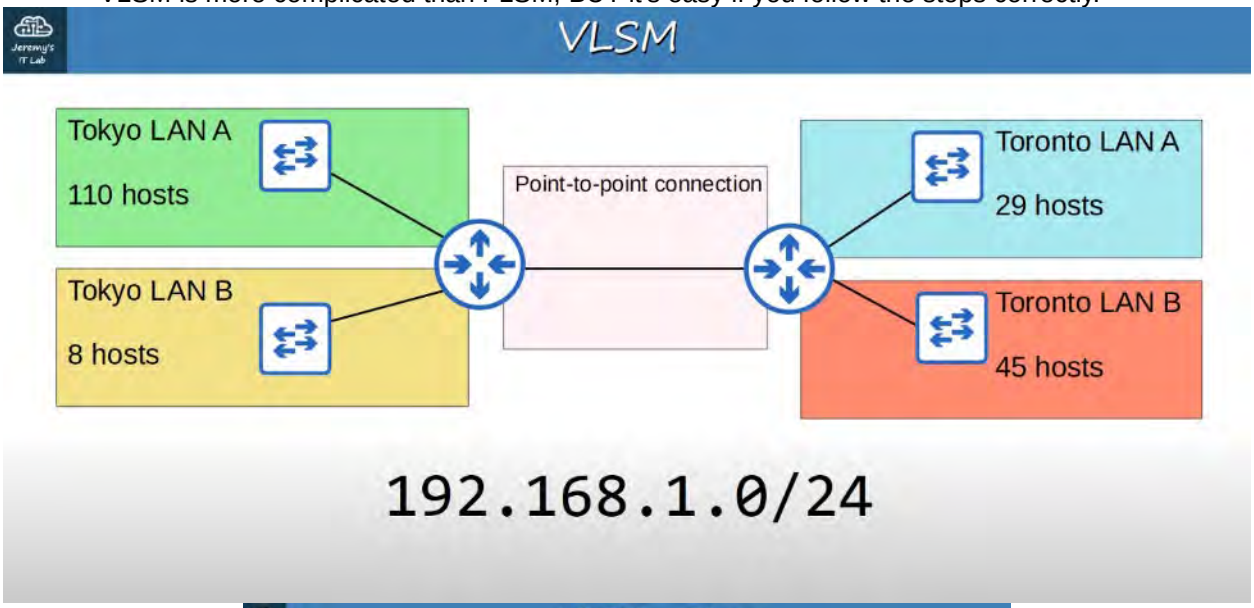
Prefix Length	Number of Subnets	Number of Hosts
/25	512	126
/26	1024	62
/27	2048	30
/28	4096	14
/29	8192	6
/30	16384	2
/31	32768	0 (2)
/32	65536	0 (1)

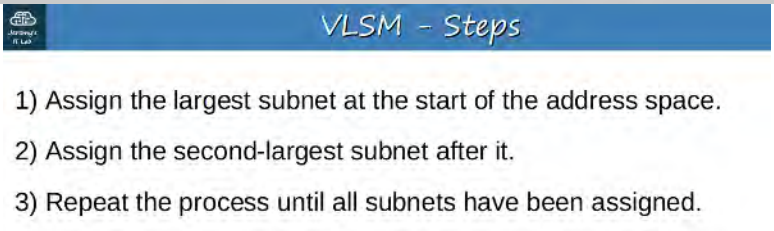
15. SUBNETTING (VLSM) : PART 3

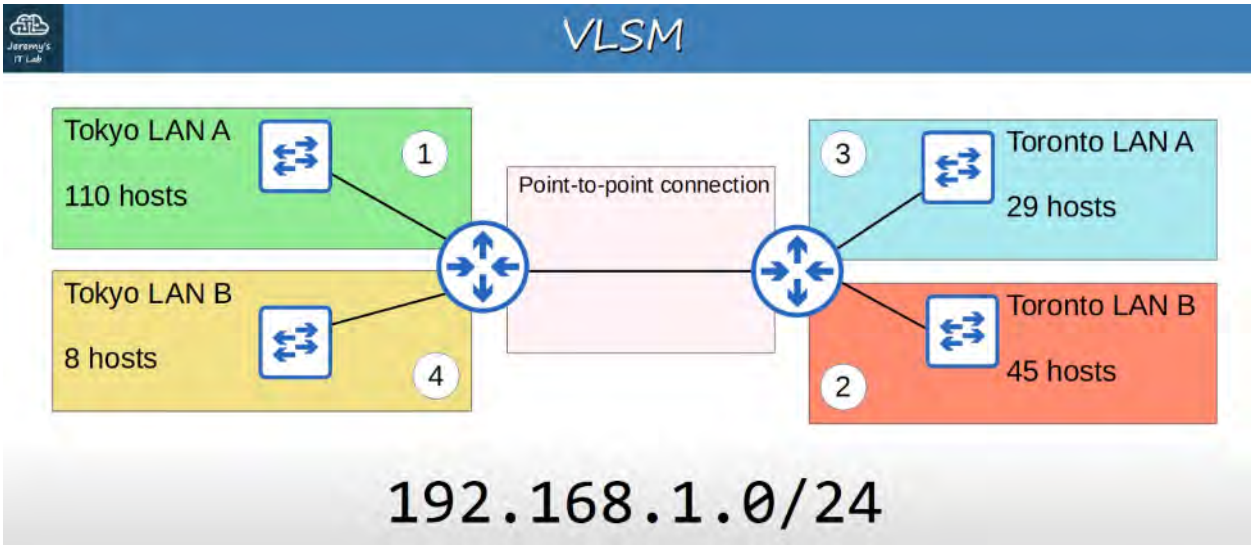
The process of subnetting Class A, Class B, and Class C is identical.
SUBNETTING CLASS A NETWORKS
Given a 10.0.0.0/8 network, you must create 2000 subnets which will distributed to various enterprises. What prefix length must you use?
2^10 = 1024 so 2^11 = 2048. We have to "borrow" 11 bits (Left to Right) to get enough subnets
0000 1010 . 0000 0000 . 000 | 00000 . 0000 0000
8 bits + 8 bits + 3 = 19 bits
0000 1010 . 0000 0000 . 000 | 00000 . 0000 0000 1111 1111 . 1111 1111 . 111 | 00000 . 0000 0000
255.255.224.0 is the Subnet mask
The answer is /19 (/8 + /11 = /19)
How many hosts per subnet? There are 13 host bits remaining so:
2^13 - 2 = 8190 hosts per subnet

VARIABLE-LENGTH SUBNET MASKS (VLSM)

- Until now, we have practiced subnetting using FLSM (Fixed-Length Subnet Masks).
- This means that all of the subnets use the same prefix length (ie: Subnetting a Class C network into 4 subnets using /26)
- VLSM (Variable-Length Subnet Masks) is the process of creating subnets of different sizes, to make your use of network addresses more efficient.
- VLSM is more complicated than FLSM, BUT it's easy if you follow the steps correctly.







So, in order:
TOKYO LAN A (110 HOSTS)
TORONTO LAN B (45 HOSTS)
TORONTO LAN A (29 HOSTS)
TOKYO LAN B (8 HOSTS)
and
THE POINT TO POINT CONNECTION (between the two ROUTERS)
192.168.1.0 / 24
1000 0000 . 1010 1000 . 0000 0001 | 0000 0000 (last is host octet = 254 usable hosts)
Shifting LEFT - we DOUBLE the # of hosts Shifting RIGHT - we HALF the # of hosts
TOKYO LAN A (we need to borrow 1 host bits, to the RIGHT, to leave enough for 2^7 or 128 hosts. More than enough for TOKYO A)
so:
192.168.1.0/25 (Network Address)
1000 0000 . 1010 1000 . 0000 0001 . 0 | 000 0000

Converting remaining Host Bits to 1s:
0111 1111, we get 127 so

192.168.1.127/25 is the Broadcast Address

TOKYO LAN A
NETWORK ADDRESS: 192.168.1.0/25
BROADCAST ADDRESS: 192.168.1.127/25
FIRST USABLE: 192.168.1.1/25
LAST USABLE: 192.168.1.126/25
TOTAL NUMBER OF USABLE HOSTS: 126 (2^7 -2)

Since TOKYO LAN A is 192.168.1.127, the next Subnet (TOKYO LAN B) starts at 192.168.1.128 (Network Address)

TORONTO LAN B
NETWORK ADDRESS: 192.168.1.128 / 26
BROADCAST ADDRESS: 192.168.1.191 / 26
FIRST USABLE: 192.168.1.129 /26
LAST USABLE: 192.168.1.190 / 26
TOTAL NUMBER OF USABLE HOSTS: 62 (2^6 -2)
We need to borrow to get enough for 45 hosts.

128	64	32	16	8	4	2	1
x	x	0	0	0	0	0	0

1000 0000 . 1010 1000 . 0000 0001 . 10 | 00 0000

192 . 168 . 1 . 128

1000 0000 . 1010 1000 . 0000 0001 . 10 | 11 1111

192 . 168 . 1 . 191 (Broadcast Address)

TORONTO LAN A
We need to borrow to get enough for 29 hosts.

128	64	32	16	8	4	2	1
x	x	x	0	0	0	0	0

1000 0000 . 1010 1000 . 0000 0001 . 110 | 0 0000

192.168.1.192 (Net Address)

1000 0000 . 1010 1000 . 0000 0001 . 110 | 1 1111

192.168.1.224 (Broadcast address)

NETWORK ADDRESS: 192.168.1.192 / 27
BROADCAST ADDRESS: 192.168.1.223 / 27
FIRST USABLE: 192.168.1.193 /27
LAST USABLE: 192.168.1.222 / 27
TOTAL NUMBER OF USABLE HOSTS: 30 hosts (2^5 - 2)

TOKYO LAN B We need to borrow to get enough for 8 hosts. Remember total usable hosts is equal to x - 2.

128	64	32	16	8	4	2	1
x	x	x	x	0	0	0	0

1000 0000 . 1010 1000 . 0000 0001 . 1110 | 0000

192.168.1.224 (Net Address)

1000 0000 . 1010 1000 . 0000 0001 . 1110 | 1111

192.168.1.239 (Broadcast address)

NETWORK ADDRESS: 192.168.1.224 / 28
BROADCAST ADDRESS: 192.168.1.239 / 28
FIRST USABLE: 192.168.1.225 /28
LAST USABLE: 192.168.1.238 / 28
TOTAL NUMBER OF USABLE HOSTS: 14 hosts (2^4 - 2)

POINT TO POINT CONNECTIONS
We need to borrow to get enough for 4 hosts. Remember total usable hosts is equal to x - 2.

128	64	32	16	8	4	2	1
x	x	x	x	x	x	0	0

1000 0000 . 1010 1000 . 0000 0001 . 1111 00 | 00

192.168.1.240 (Net Address)

1000 0000 . 1010 1000 . 0000 0001 . 1111 00 | 11

192.168.1.243 (Broadcast address)

NETWORK ADDRESS: 192.168.1.240 / 30
BROADCAST ADDRESS: 192.168.1.243 / 30
FIRST USABLE: 192.168.1.241 / 30
LAST USABLE: 192.168.1.242 / 30
TOTAL NUMBER OF USABLE HOSTS: 2 hosts (2^2 - 2)

ADDITIONAL PRACTICE FOR SUBNETTING
http://www.subnettingquestions.com http://subnetting.org https://subnettingpractice.com *** Preferred site ***

16. VLANS : PART 1

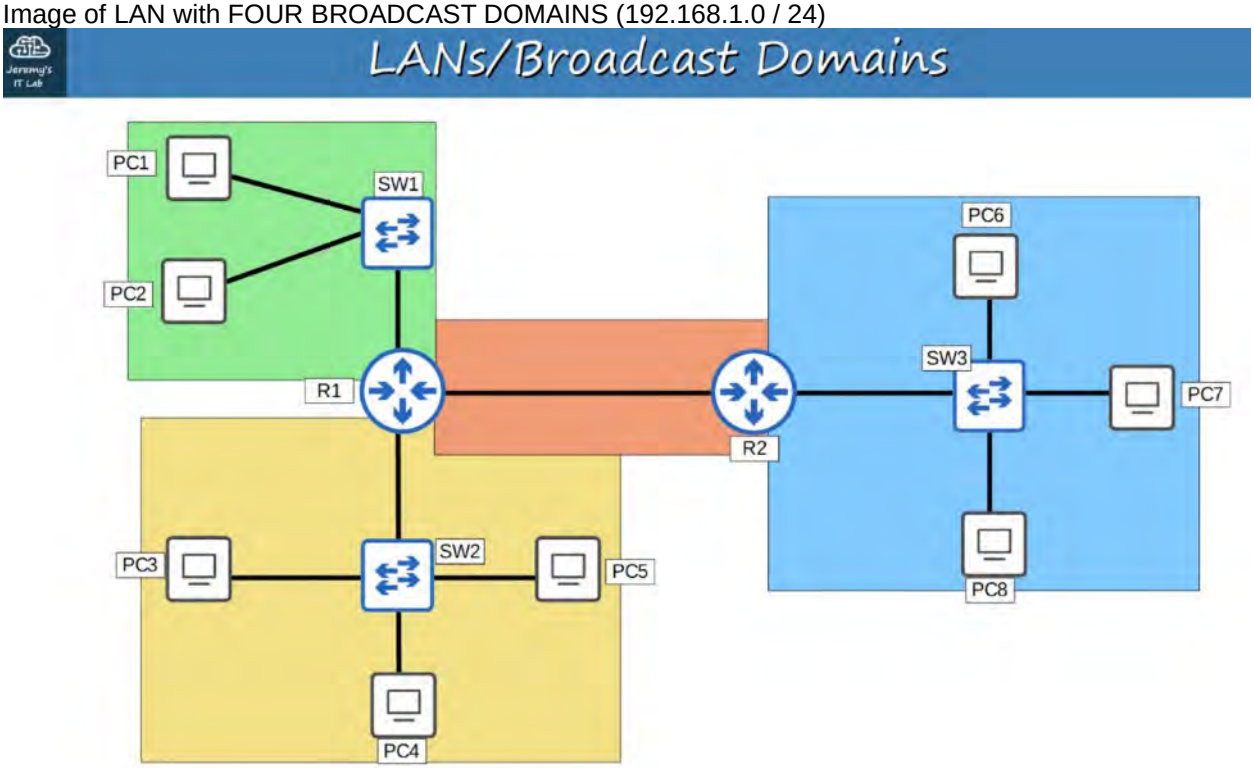
WHAT IS A LAN ?

- A LAN is a single BROADCAST DOMAIN, including all devices in that broadcast domain.

BROADCAST DOMAINS

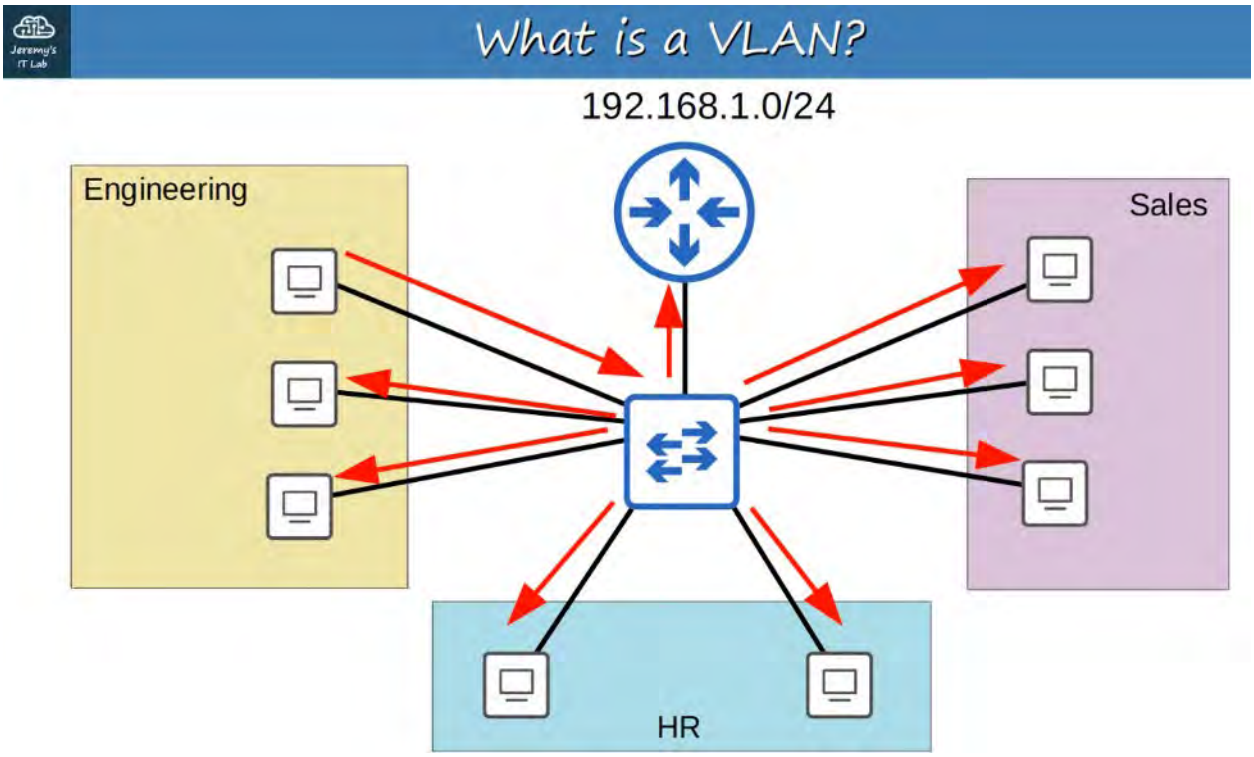
- A BROADCAST DOMAIN is the group of devices which will receive a BROADCAST FRAME (Destination MAC : FFFF.FFFF.FFFF) sent by any one of the members.

Image of LAN with FOUR BROADCAST DOMAINS (192.168.1.0 / 24)

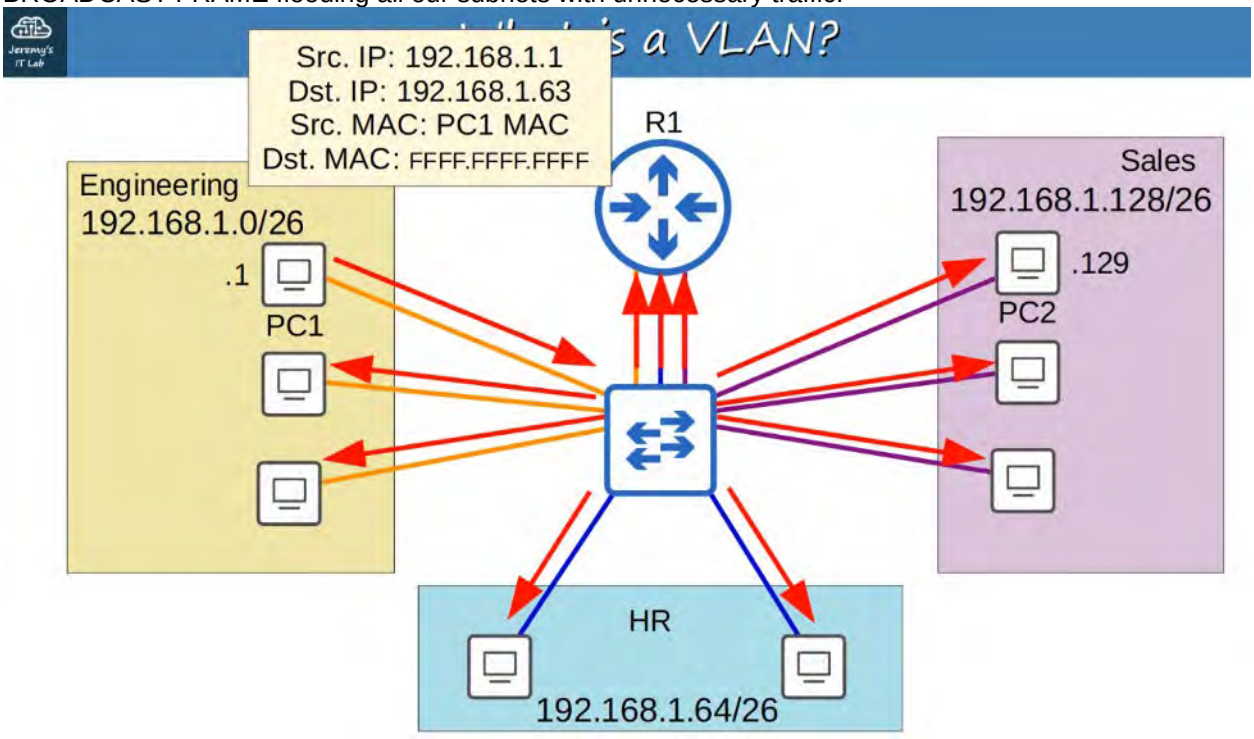


Performance :

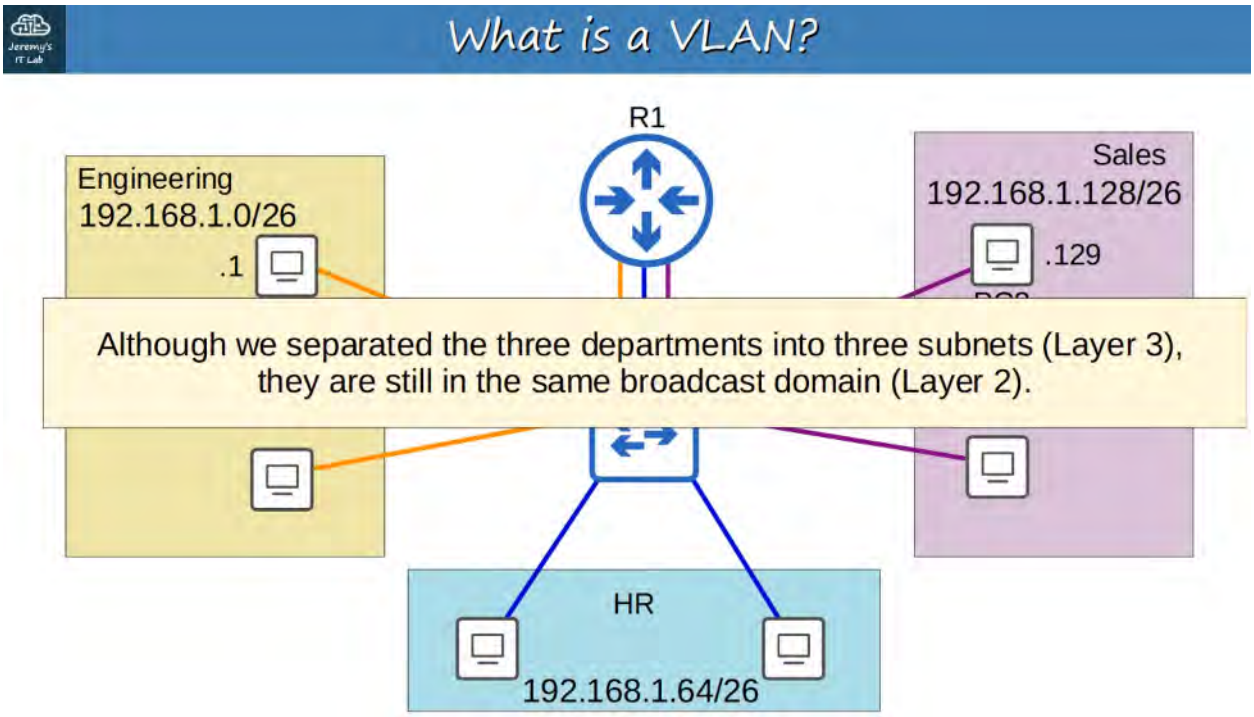
Lots of unnecessary BROADCAST traffic can reduce network performance.



BROADCAST FRAME flooding all our subnets with unnecessary traffic.



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, PC's can reach each other directly, without traffic passing through the router. So, even if you configure security policies, they won't have any effect.



WHAT IS A VLAN ?
VLANS:

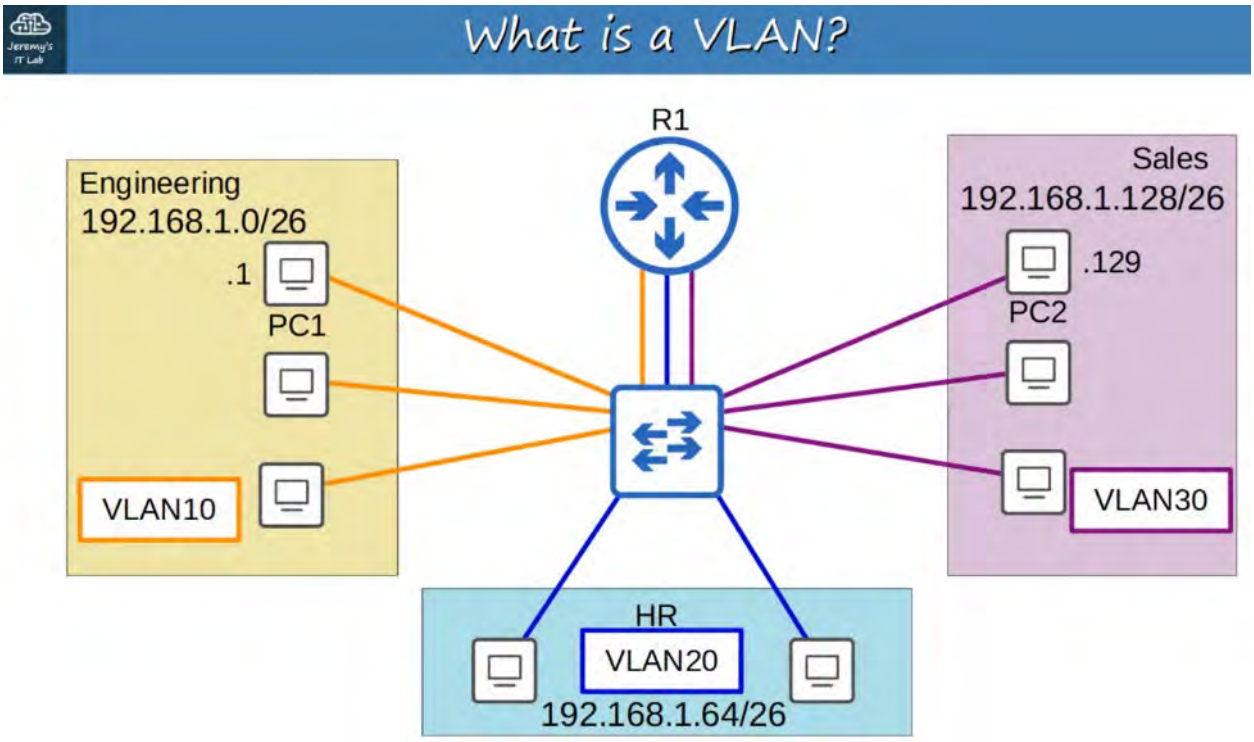
- logically separate end-hosts at LAYER 2
- are configured on Layer 2 SWITCHES on a per-interface basis.
- any END HOST connected to that interface is part of that VLAN

PURPOSE OF VLANs:
Network Performance :

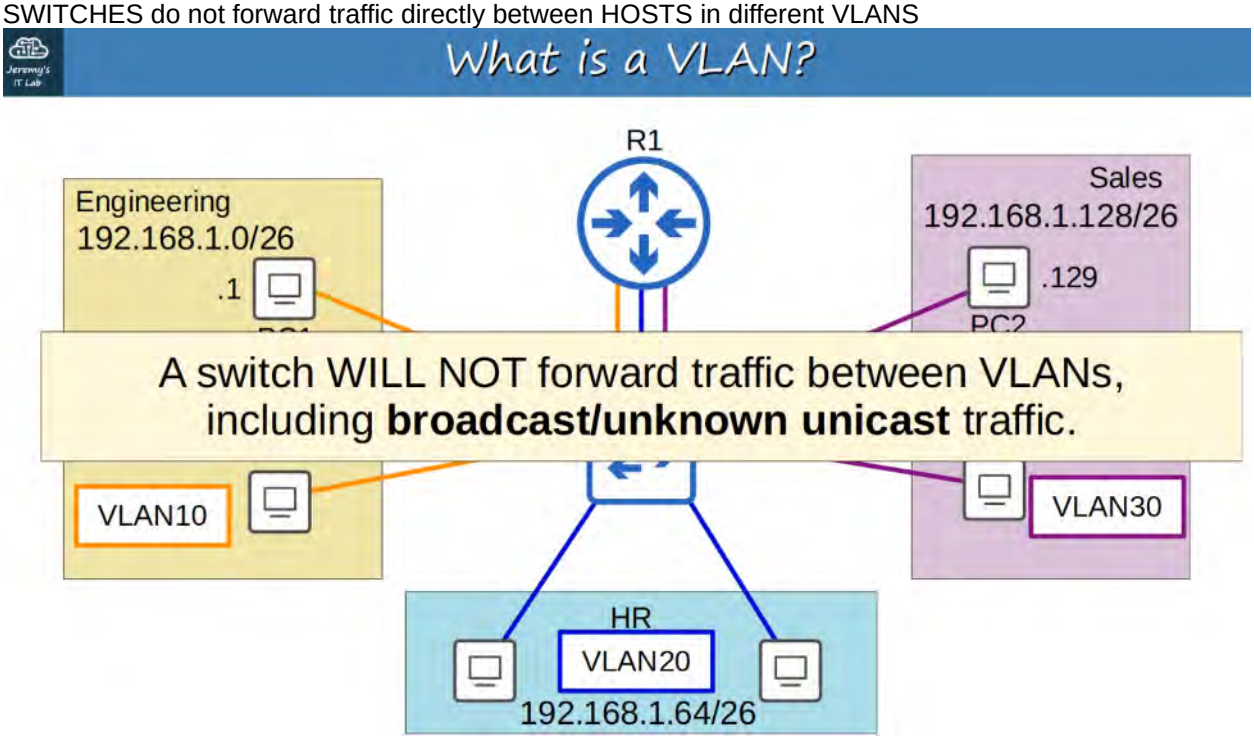
- Reduce unnecessary BROADCAST traffic, which helps prevent network congestion, and improve network performance

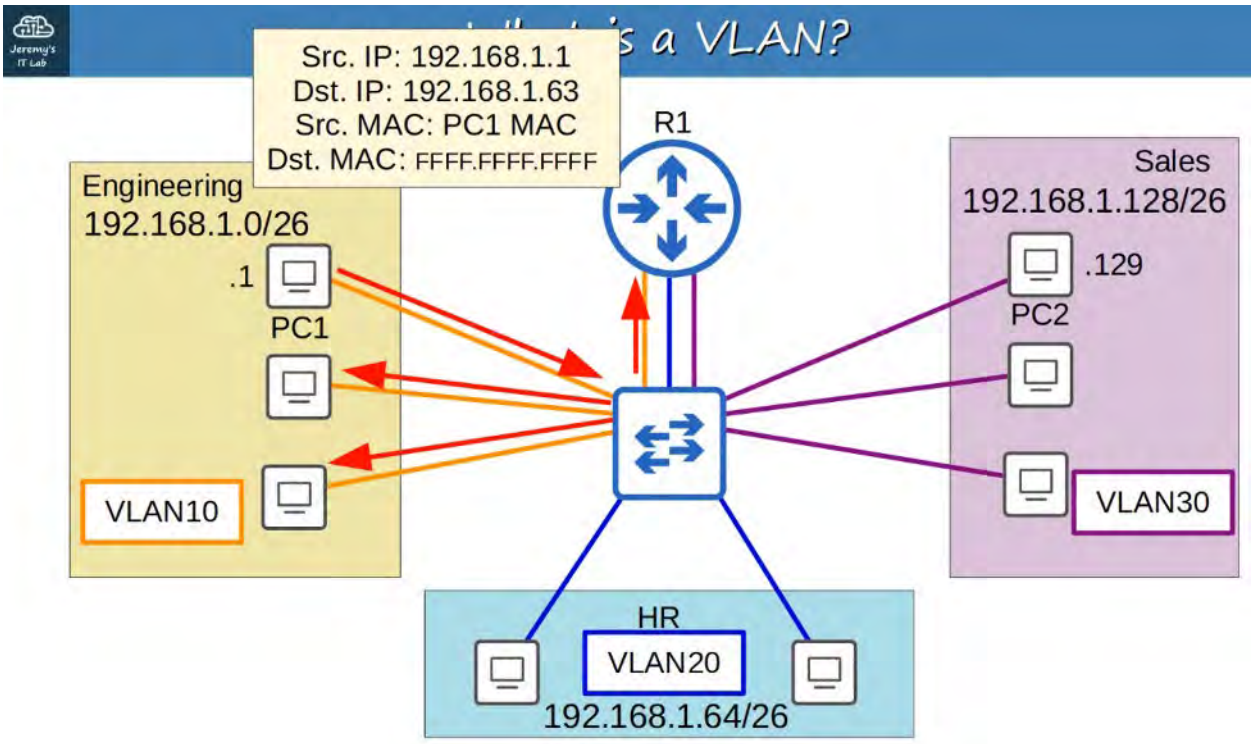
Network Security :

- Limiting BROADCAST and unknown UNICAST traffic, also improves network security, since messages won't be received by devices outside of the VLAN

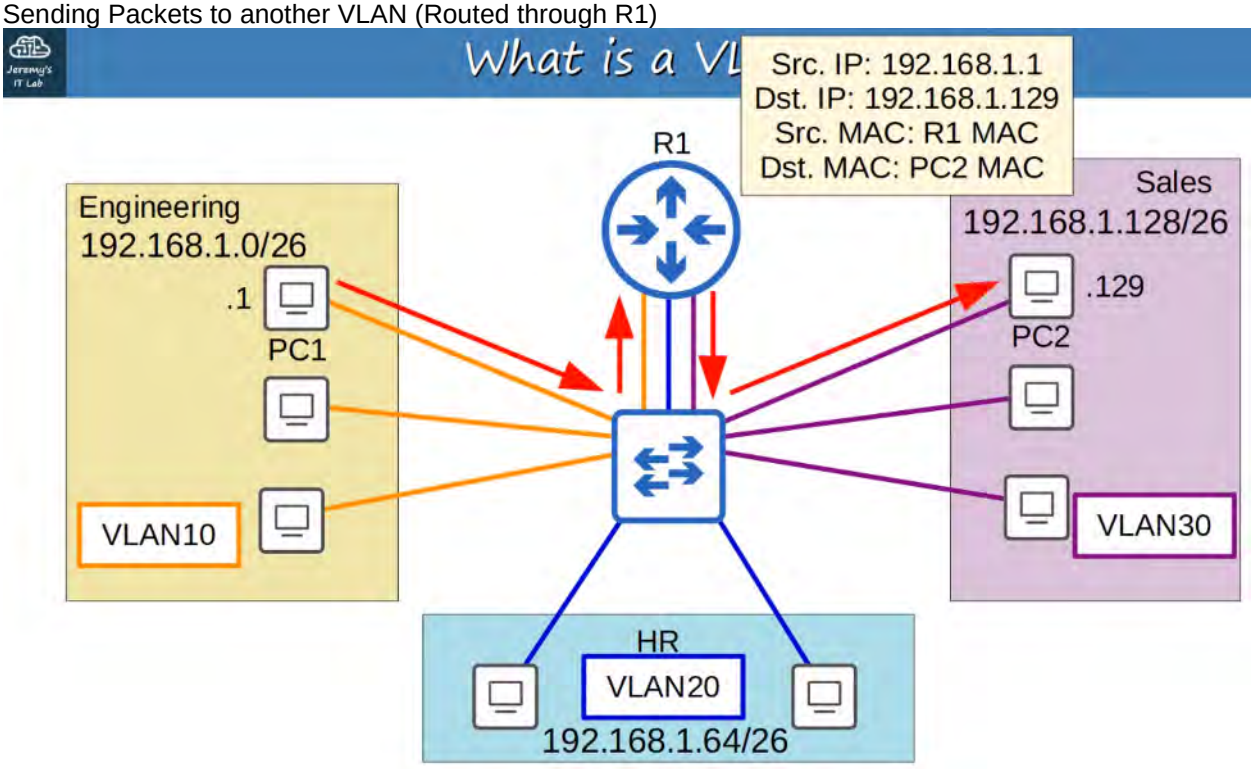


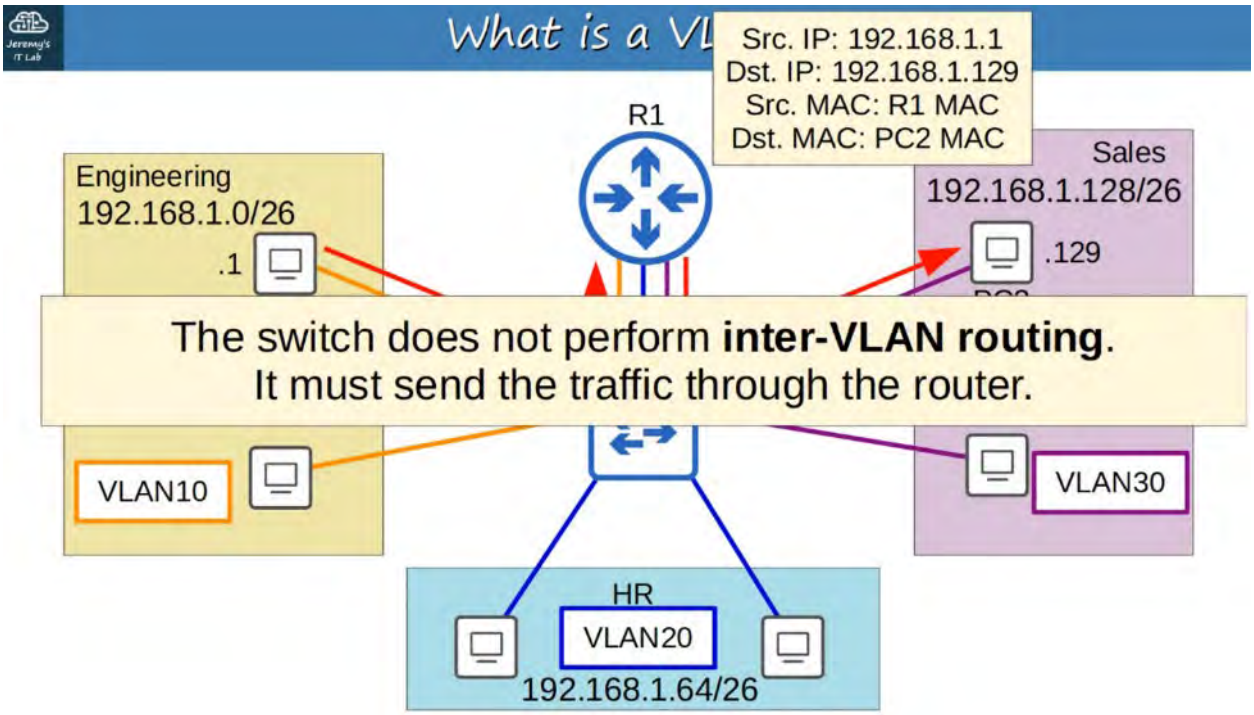
SWITCHES do not forward traffic directly between HOSTS in different VLANS



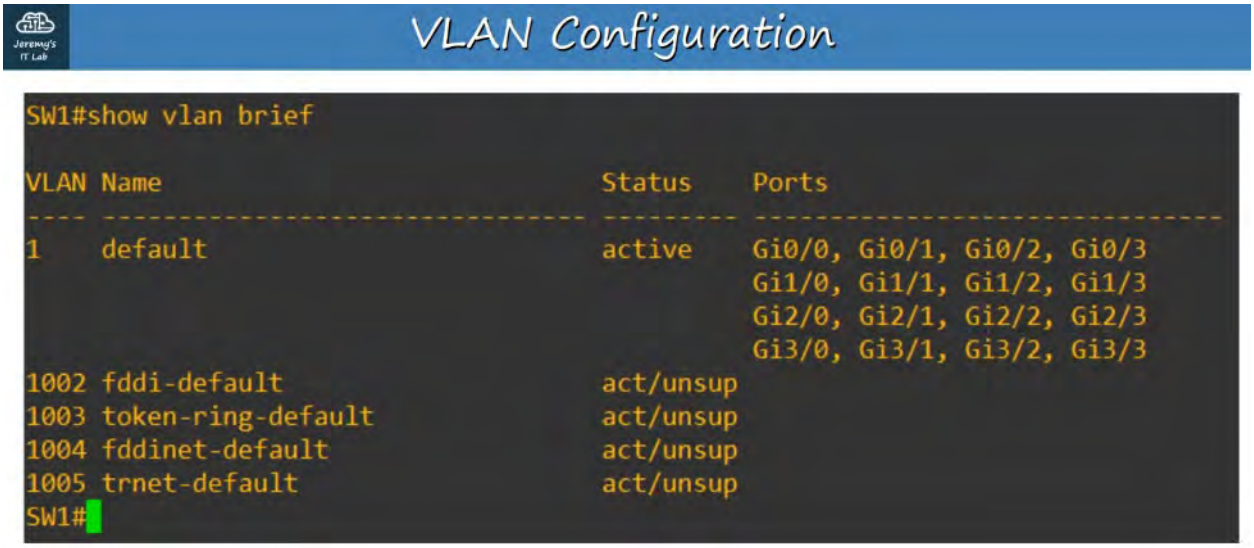


Sending Packets to another VLAN (Routed through R1)





HOW TO CONFIGURE VLANS ON CISCO SWITCHES
#show vlan brief



Shows which VLANS that exist on the SWITCH and what INTERFACES are in each VLAN
VLANs 1 (DEFAULT), 1002-1005 exist by default and **cannot be deleted (5 VLANs)**

HOW TO ASSIGN INTERFACES TO A VLAN

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)#
```

1. Use the “interface range” command to select all the interfaces at once
2. Use the “switchport mode access” command to set the interface as an ACCESS PORT

WHAT IS AN ACCESS PORT?

- 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 info on TRUNK in next chapter)

3. Use the “switchport access” command to assign a VLAN to a PORT

VLAN Configuration

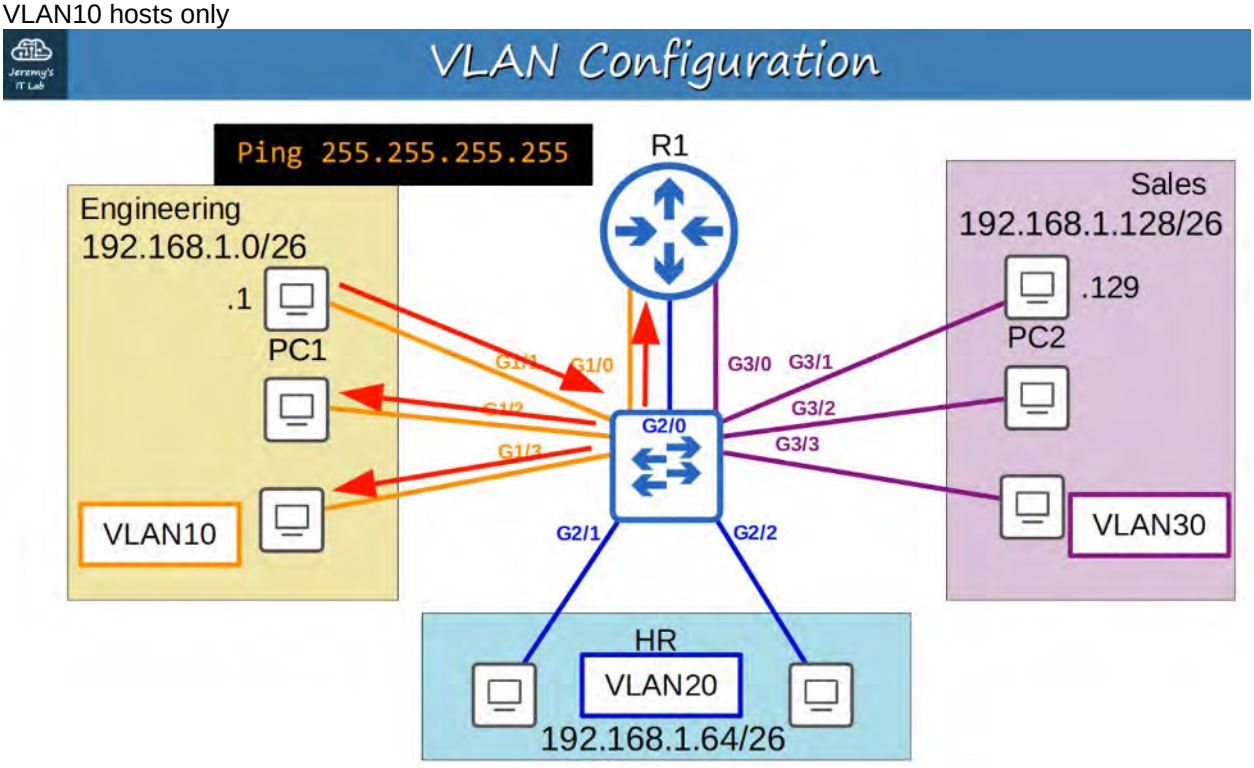
```
SW1(config)#do show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi0/0, Gi0/1, Gi0/2, Gi0/3
                                                Gi2/3
10   VLAN0010                         active    Gi1/0, Gi1/1, Gi1/2, Gi1/3
20   VLAN0020                         active    Gi2/0, Gi2/1, Gi2/2
30   VLAN0030                         active    Gi3/0, Gi3/1, Gi3/2, Gi3/3
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
SW1(config)#vlan 10
SW1(config-vlan)#name ENGINEERING
SW1(config-vlan)#vlan 20
SW1(config-vlan)#name HR
SW1(config-vlan)#vlan 30
SW1(config-vlan)#name SALES
```

Use “#vlan <#>” to enter **Configuration Mode** for a given VLAN (this can also create a VLAN)
Use “#name ” to configure a NAME for your VLAN
To check your VLAN configuration, use “#show vlan brief”

VLAN Configuration

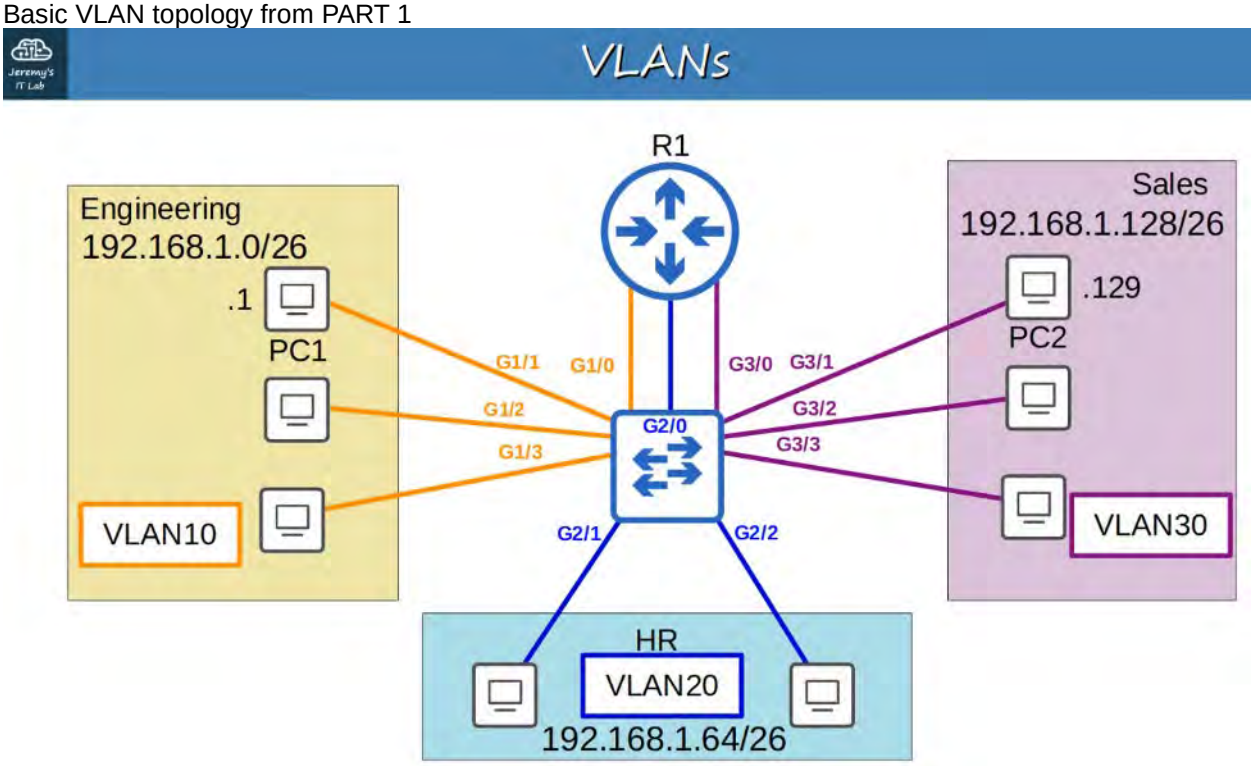
```
SW1(config)#do show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi0/0, Gi0/1, Gi0/2, Gi0/3
                                                Gi2/3
10   ENGINEERING                      active    Gi1/0, Gi1/1, Gi1/2, Gi1/3
20   HR                               active    Gi2/0, Gi2/1, Gi2/2
30   SALES                            active    Gi3/0, Gi3/1, Gi3/2, Gi3/3
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
SW1(config)#
```

Testing VLAN 10
Pinging from PC1 using 255.255.255.255 (FFFF:FFFF:FFFF) floods broadcast packets to R1 and VLAN10 hosts only

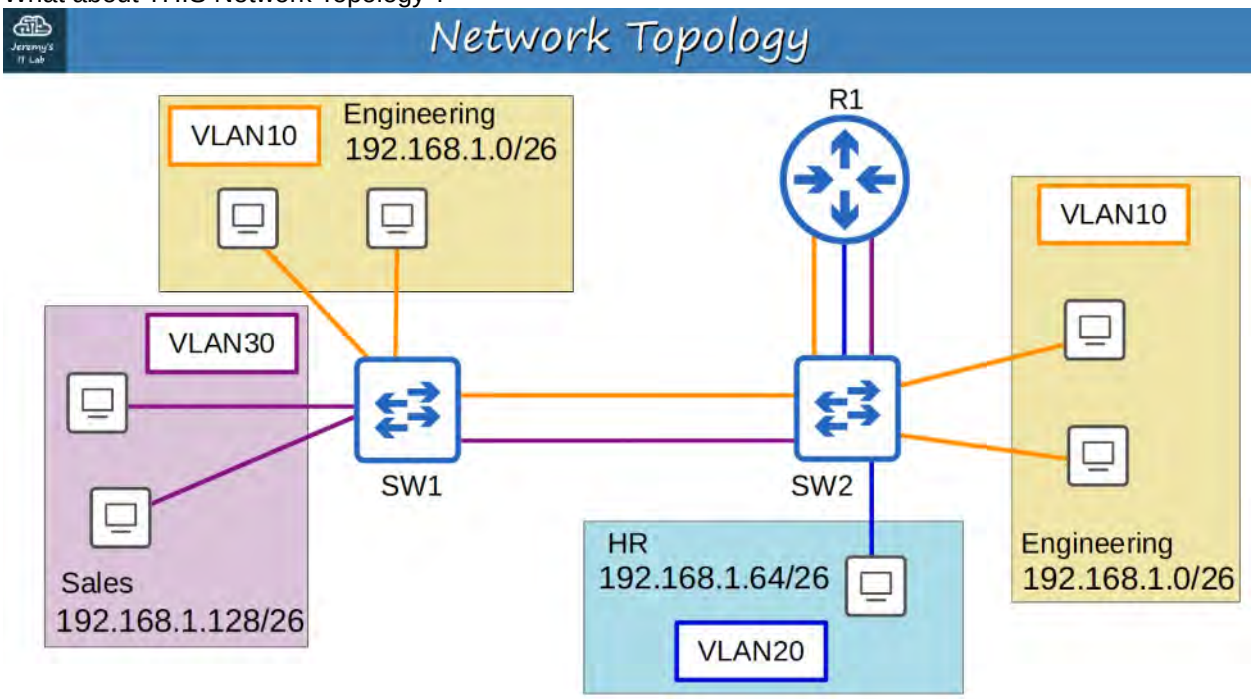


17. VLANS : PART 2

Basic VLAN topology from PART 1



What about THIS Network Topology ?

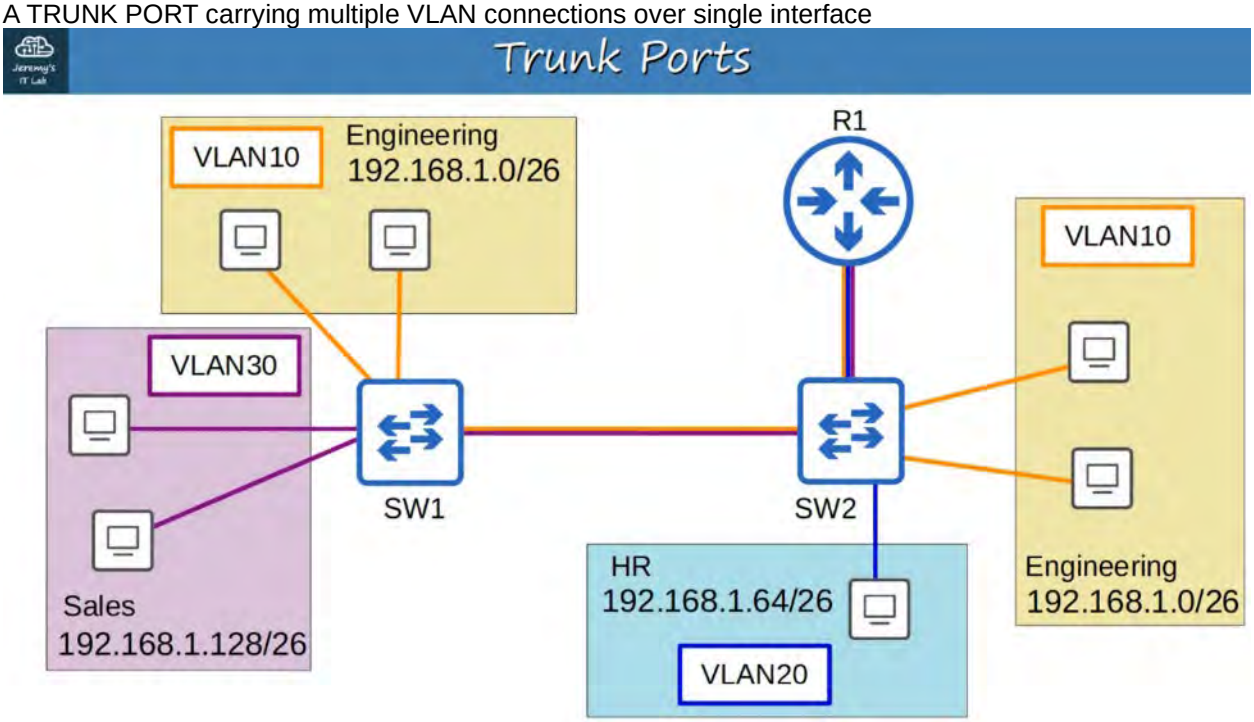


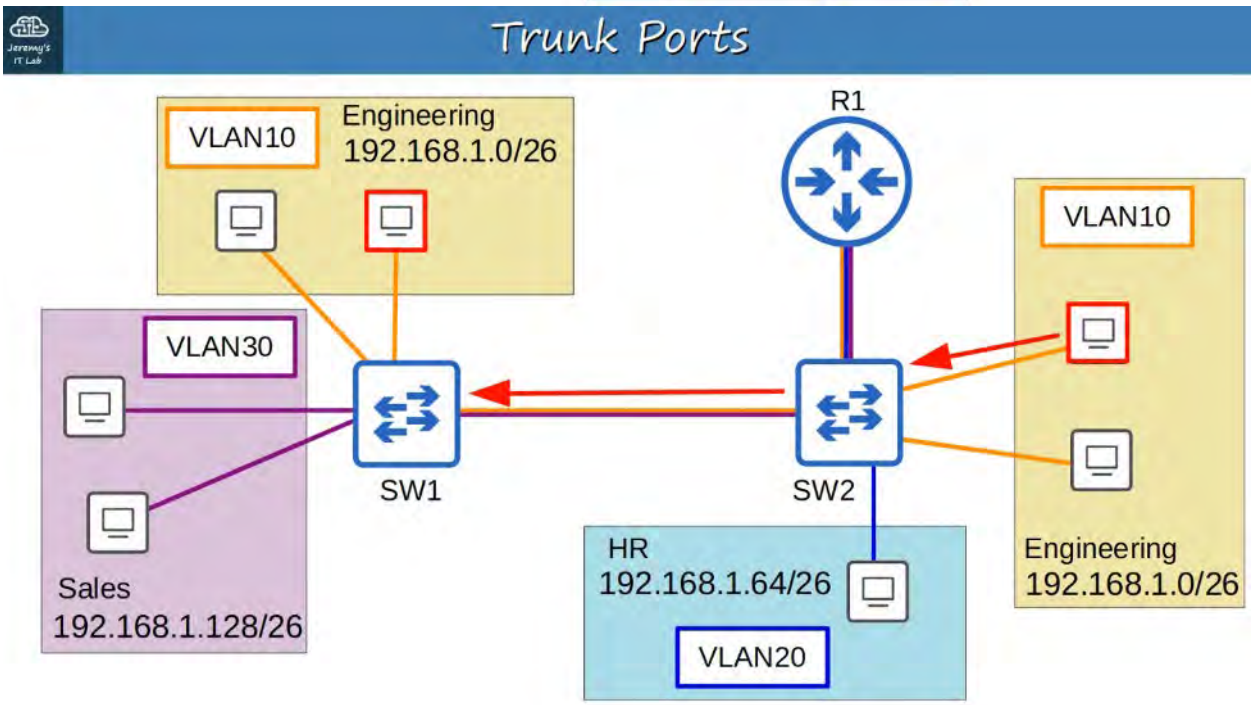
Notice this one has TWO Switches (SW1 and SW2) and ENGINEERING (VLAN 10) has two separate locations on the network.

TRUNK PORTS

- In a small network with few VLANS, it's possible to use a separate interface for EACH VLAN when connecting SWITCHES to SWITCHES, and SWITCHES to ROUTERS
- HOWEVER, when the number of VLANS increases, this is not viable. It will result in wasted interfaces, and often ROUTERS won't have enough INTERFACES for each VLAN
- You can use TRUNK PORTS to carry traffic from multiple VLANS over a single interface

A TRUNK PORT carrying multiple VLAN connections over single interface





How does a packet know WHICH VLAN to send traffic to over the TRUNK PORT ?
VLAN TAGS !
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.

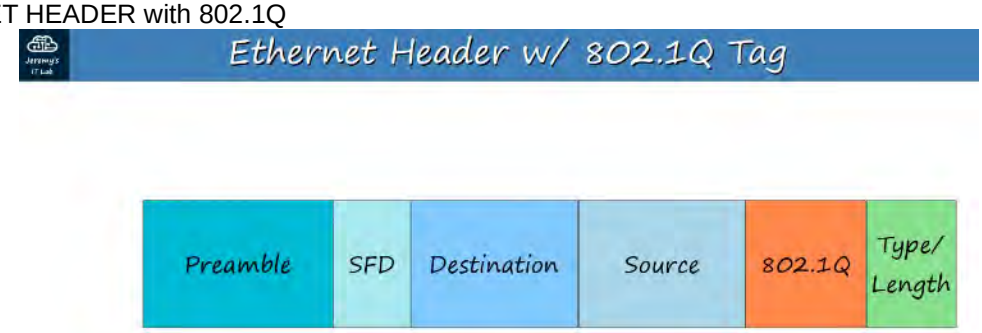
TRUNK PORT = “Tagged” ports
ACCESS PORT = “Untagged” ports

VLAN TAGGING

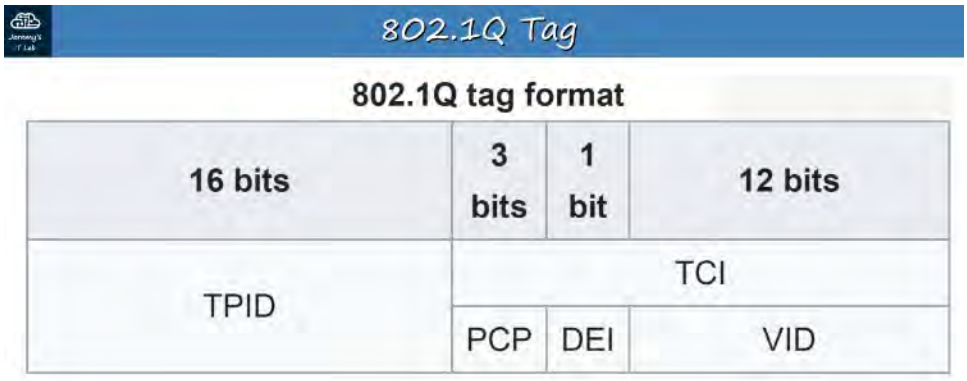
- There are TWO main TRUNK protocols:
  - ISL (Inter-Switch Link)
  - IEEE 802.1Q (also known as “dot1q”)

ISL is an old Cisco proprietary protocol created before 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 use it.
For the CCNA, you will only need to learn 802.1Q

ETHERNET HEADER with 802.1Q



- The 802.1Q TAG Is inserted between the SOURCE and TYPE/LENGTH fields in 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)
    - TCI consists of THREE sub-fields:



TPID (TAG Protocol Identifier) :

- 16 bits (2 bytes) in length
- Always set to a value of 0x8100. This indicates that the frame is 802.1Q TAG

TCI / PCP (Priority Code Point) :

- 3 bits in length
- Used for Class of Service (CoS), which prioritizes important traffic in congested networks

TCI / DEI (Drop Eligible Indicator) :

- 1 bit in length
- Used to indicated frames that can be dropped if the network is congested

TCI / VID (VLAN ID) :

- 12 bits in length

- Identifies the VLAN the frame belongs to
- 12 bits in length = 4096 total VLANS (2^12), range of 0 - 4095
- VLANs 0 and 4095 are reserved and can't be used
- Therefore, the actual range of VLANs is 1 - 4094

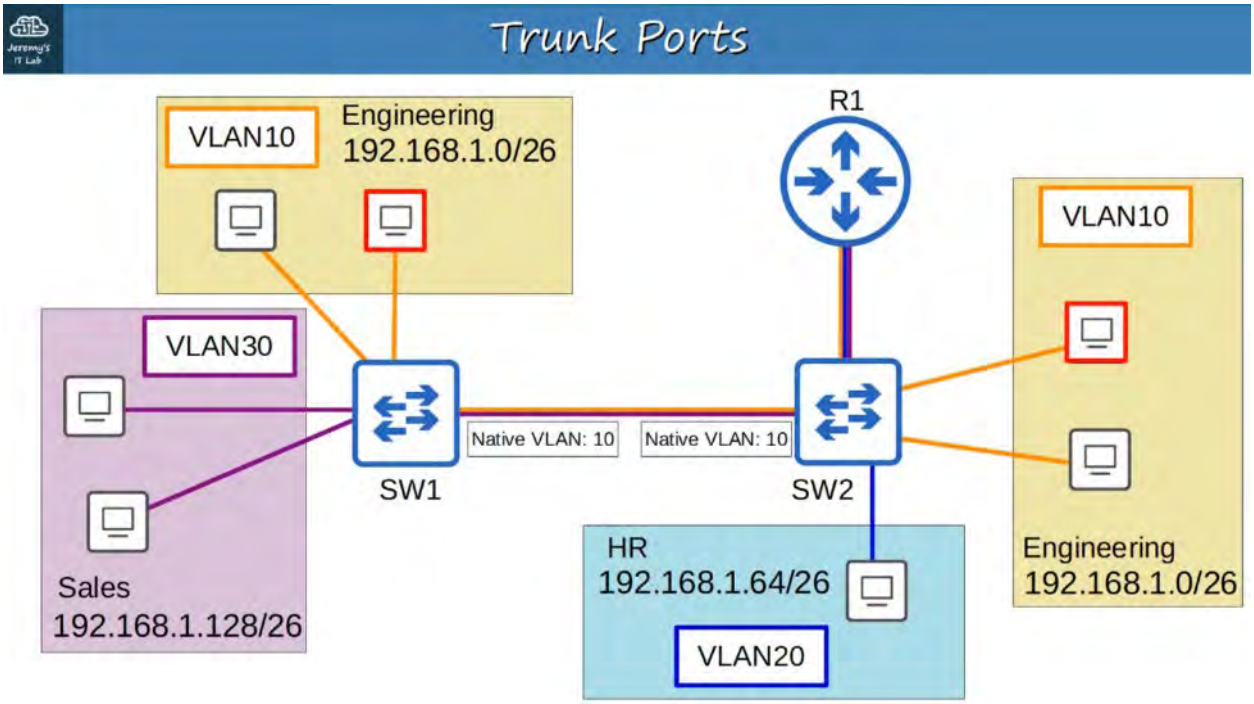
NOTE : Cisco's ISL also had a VLAN range of 1 - 4094

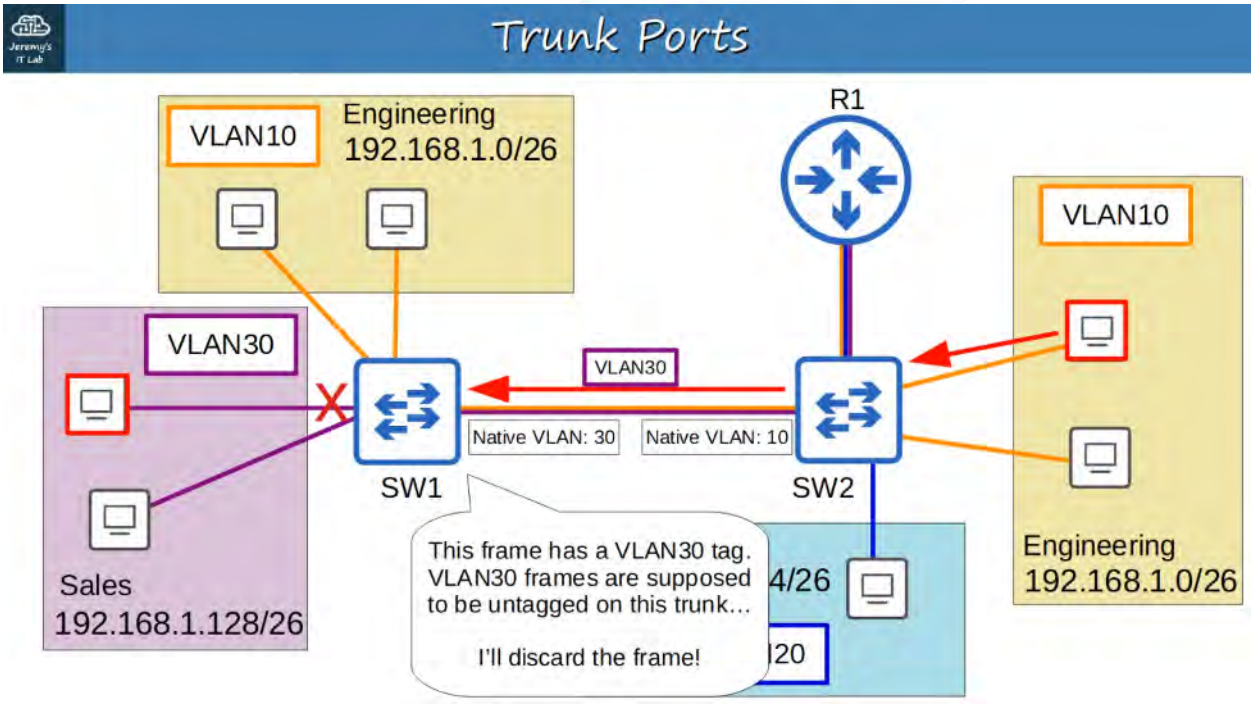
VLAN RANGES

- The range of VLANs (1 – 4094) is divided into two sections:
  Normal VLANs: 1 – 1005
  Extended VLANs: 1006 – 4094
- Some older devices cannot use the extended VLAN range, however it's safe to expect that modern switches will support the extended VLAN range.

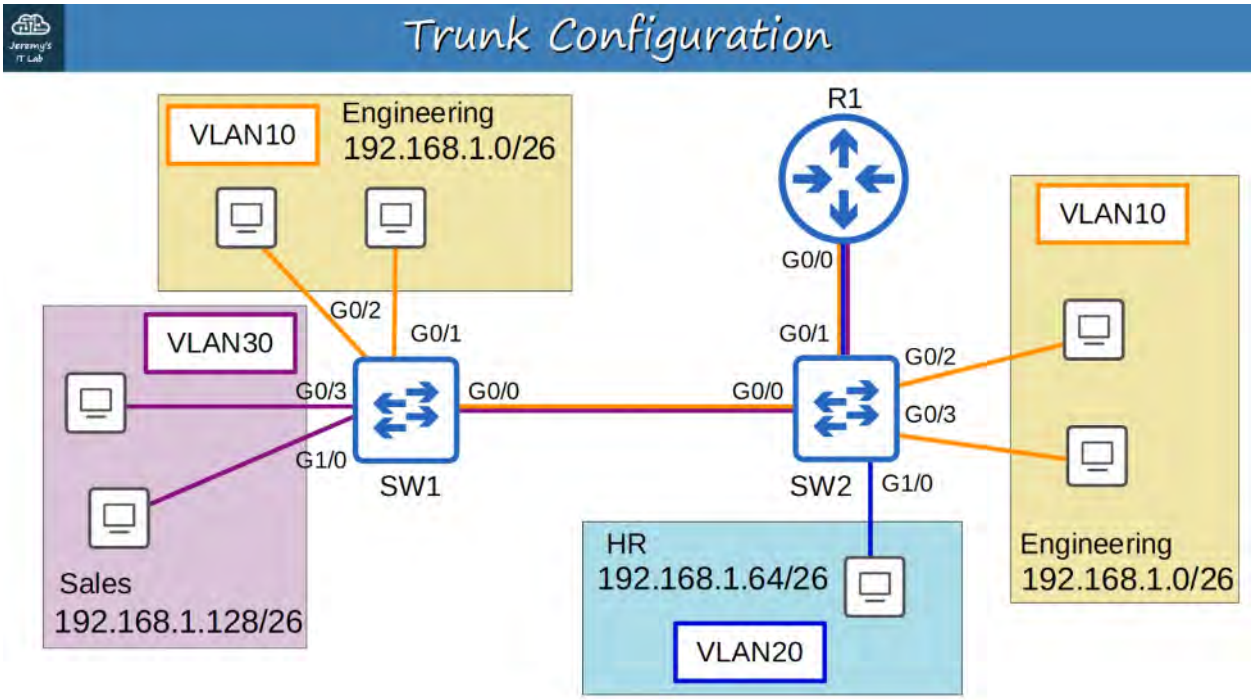
NATIVE VLAN

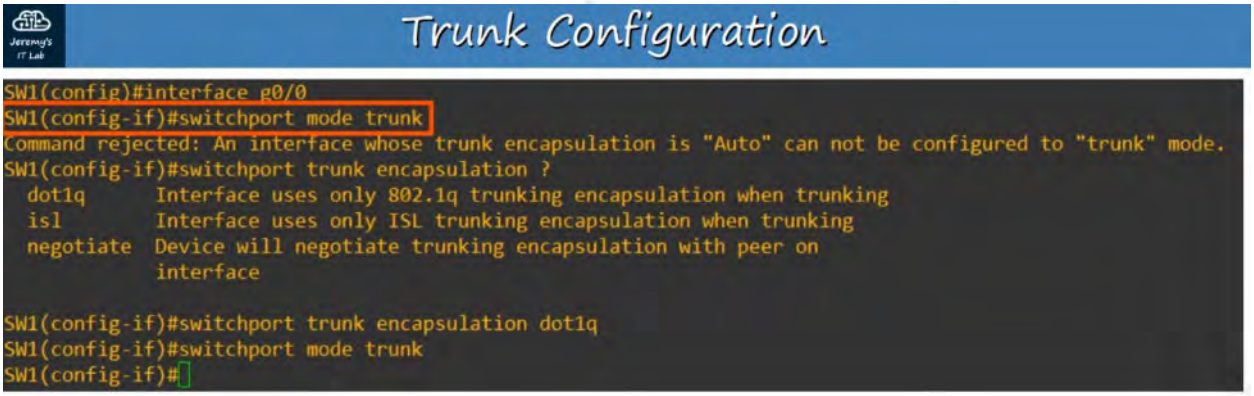
- 802.1Q has a feature called the **native VLAN**. (ISL does not have this feature)
- The native VLAN is VLAN 1 by default on all trunk ports, however this can be manually configured on each trunk port.
- The switch does not add an 802.1Q tag to frames in the native VLAN.
- When a switch receives an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





TRUNK CONFIGURATION





Many modern switches do not support Cisco's ISL at all. They only support 802.1Q (dot1q)
However, SWITCHES that do support both (like the one I am using in this example) have a TRUNK encapsulation of "AUTO" by default
To MANUALLY configure the INTERFACE as a TRUNK PORT, you must first set the encapsulation to "802.1Q" or "ISL". On SWITCHES that only support 802.1Q, this is not necessary
After you set the encapsulation type, you can then configure the interface as a TRUNK

1. Select the interface to configure
2. Use "#switchport trunk encapsulation dot1q" to set the encapsulation mode to 802.1Q
3. Use "#switchport mode trunk" to manually configure the interface to TRUNK

Trunk Configuration

```
SW1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1,10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       1,10,30
SW1#
```

Use the “#show interfaces trunk” command to confirm INTERFACES on TRUNK

Trunk Configuration

```
SW1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1,10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       1,10,30
SW1#
```

```
SW1#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Gi1/1, Gi1/2, Gi1/3, Gi2/0
                                                Gi2/1, Gi2/2, Gi2/3, Gi3/0
                                                Gi3/1, Gi3/2, Gi3/3
10   ENGINEERING                      active    Gi0/1, Gi0/2
30   SALES                            active    Gi0/3, Gi1/0
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
SW1#
```

Commands to allow a VLAN on a given TRUNK

Trunk Configuration

```
SW1(config)#int g0/0
SW1(config-if)#
SW1(config-if)#switchport trunk allowed vlan ?
  WORD    VLAN IDs of the allowed VLANs when this port is in trunking mode
  add     add VLANs to the current list
  all     all VLANs
  except  all VLANs except the following
  none    no VLANs
  remove  remove VLANs from the current list

SW1(config-if)#switchport trunk allowed vlan
```

Trunk Configuration

```
SW1(config-if)#switchport trunk allowed vlan 10,30
SW1(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1

Port        Vlans allowed and active in management domain
Gi0/0       10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
SW1(config-if)#
```

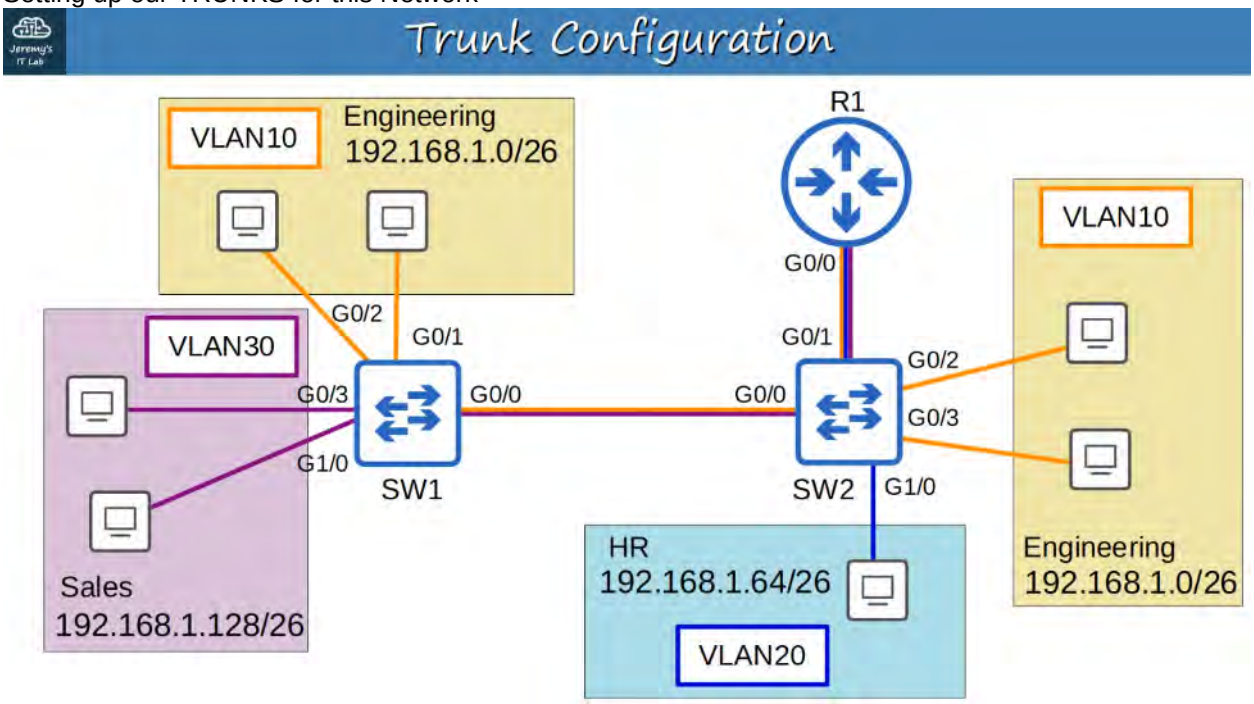
For security purposes, it is best to change the native VLAN to an **unused VLAN**.
(network security will be explained more in-depth later in the course)
Make sure the native VLAN matches on between switches

Command to change the NATIVE VLAN

Trunk Configuration

```
SW1(config-if)#switchport trunk native vlan 1001
SW1(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1001

Port        Vlans allowed on trunk
Gi0/0       10,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
SW1(config-if)#
```

Setting up our TRUNKS for this Network



We will need to configure :
SW1 : g0/0 interface (already configure above this section)
SW2: g0/0, and g0/1 interface
SW2 g0/0

Trunk Configuration

```
SW2(config)#interface g0/0
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 10,30
SW2(config-if)#switchport trunk native vlan 1001
SW2(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1001

Port        Vlans allowed on trunk
Gi0/0       10,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
SW2(config-if)#
```

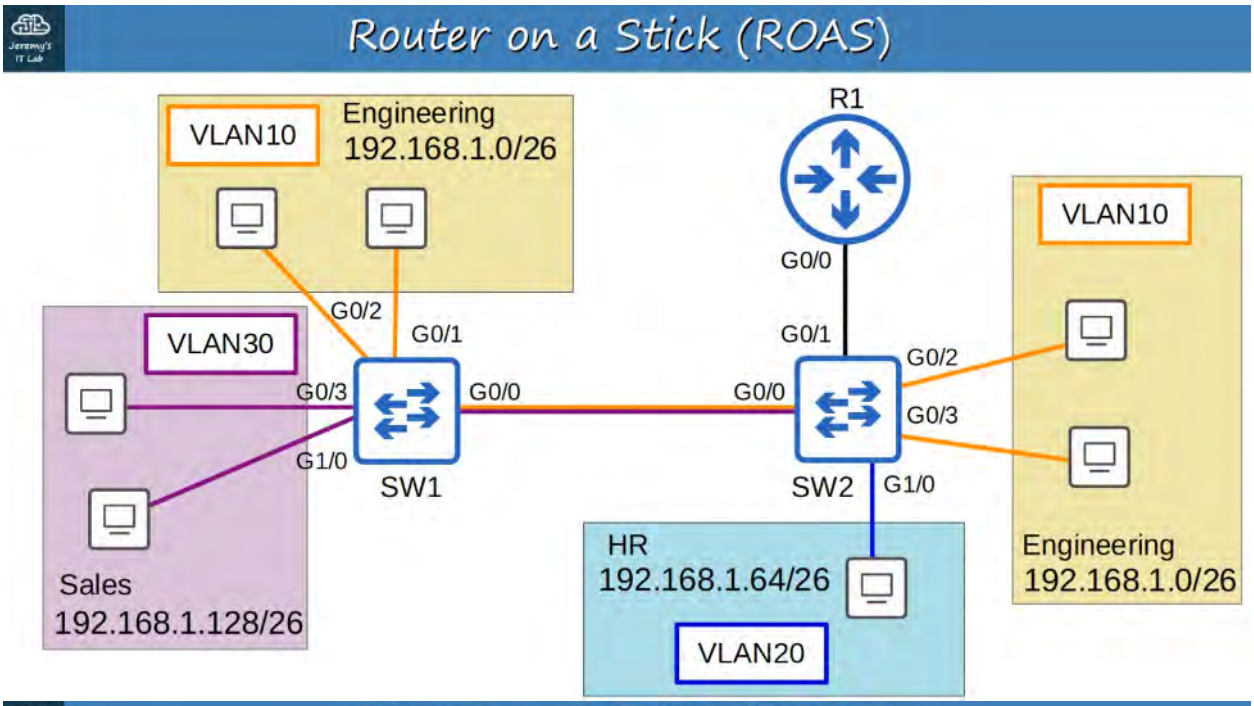
SW2 g0/1

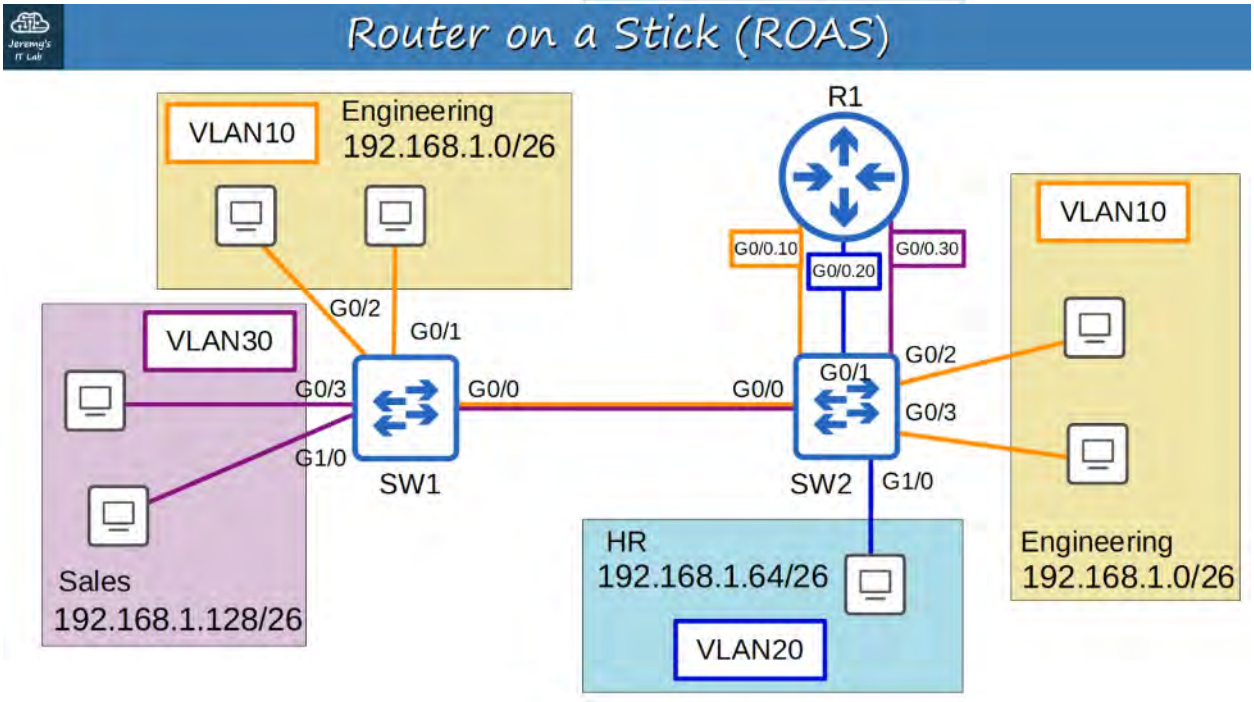
Trunk Configuration

```
SW2(config)#interface g0/1
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 10,20,30
SW2(config-if)#switchport trunk native vlan 1001
SW2(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1001
Gi0/1       on               802.1q         trunking      1001

Port        Vlans allowed on trunk
Gi0/0       10,30
Gi0/1       10,20,30

Port        Vlans allowed and active in management domain
Gi0/0       10,30
Gi0/1       10,20,30

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10,30
Gi0/1       none
SW2(config-if)#
```

What about the ROUTER, R1 ?

ROUTER ON A STICK (ROAS)


Router on a Stick (ROAS)
VLAN10
Engineering
192.168.1.0/26
R1
G0/0
VLAN10
G0/2
G0/1
G0/1
G0/2
VLAN30
G0/3
G0/0
G0/0
G0/3
G1/0
SW1
SW2
G1/0
HR
192.168.1.64/26
Engineering
192.168.1.0/26
Sales
192.168.1.128/26
VLAN20


Router on a Stick (ROAS)
VLAN10
Engineering
192.168.1.0/26
R1
G0/0.10
G0/0.20
G0/0.30
VLAN10
G0/2
G0/1
G0/1
G0/2
VLAN30
G0/3
G0/0
G0/0
G0/3
G1/0
SW1
SW2
G1/0
HR
192.168.1.64/26
Engineering
192.168.1.0/26
Sales
192.168.1.128/26
VLAN20

Router on a Stick (ROAS)

```
R1(config)#interface g0/0
R1(config-if)#no shutdown
R1(config-if)#
*Apr 15 04:29:49.681: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Apr 15 04:29:50.682: %LINEPROTO-5-UPDOWN: 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)#
```

NOTE the Sub-Interface names (like the network diagram) of 0.10, 0.20 and 0.30
You assign them IP addresses identically like you would a regular interface (using the last usable IP address of a given VLAN subnet)
Sub-interfaces will appear with the “show ip interface brief” command

Router on a Stick (ROAS)

```
R1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES NVRAM  up                    up
GigabitEthernet0/0.10  192.168.1.62    YES manual up                    up
GigabitEthernet0/0.20  192.168.1.126   YES manual up                    up
GigabitEthernet0/0.30  192.168.1.190   YES manual up                    up
GigabitEthernet0/1     unassigned      YES NVRAM  administratively down down
GigabitEthernet0/2     unassigned      YES NVRAM  administratively down down
GigabitEthernet0/3     unassigned      YES NVRAM  administratively down down
```

They also appear in the “show ip route” command (Route Table)

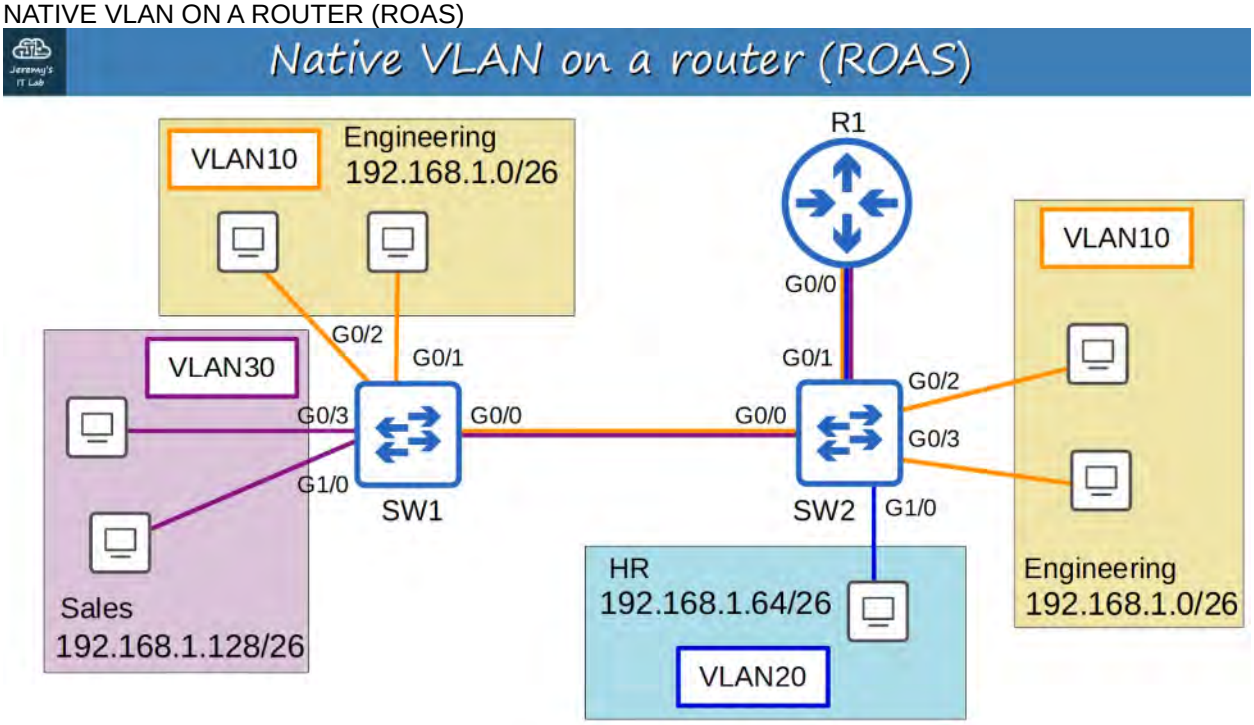
Router on a Stick (ROAS)

```
R1#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
       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, 6 subnets, 2 masks
C        192.168.1.0/26 is directly connected, GigabitEthernet0/0.10
L        192.168.1.62/32 is directly connected, GigabitEthernet0/0.10
C        192.168.1.64/26 is directly connected, GigabitEthernet0/0.20
L        192.168.1.126/32 is directly connected, GigabitEthernet0/0.20
C        192.168.1.128/26 is directly connected, GigabitEthernet0/0.30
L        192.168.1.190/32 is directly connected, GigabitEthernet0/0.30
R1#
```

ROAS is used to route between multiple VLANs using a SINGLE interface on a ROUTER and SWITCH
The SWITCH interface is configured as a regular TRUNK
The ROUTER interface is configured using SUB-INTERFACES. You configure the VLAN tag and IP address on EACH SUB-INTERFACE
The ROUTER will behave as if frames arriving with a certain VLAN tag have arrived on the SUB-INTERFACE configured with that VLAN tag
The ROUTER will TAG frames sent out of EACH SUB-INTERFACE with the VLAN TAG configured on the SUB-INTERFACE

18. VLANS : PART 3

NATIVE VLAN ON A ROUTER (ROAS)



Native VLAN untagged frames are faster and more efficient (smaller) than tagged ones.
Let's reset all SWITCHES (SW1 and SW2) to native vlan 10

```
SW1(config)#int g0/0
SW1(config-if)#switchport trunk native vlan 10
SW1(config-if)#
```

```
SW2(config)#int g0/0
SW2(config-if)#switchport trunk native vlan 10
SW2(config-if)#int g0/1
SW2(config-if)#switchport trunk native vlan 10
SW2(config-if)#
```

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

- Use the command "encapsulation dot1q " on a Sub-Interface

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

OR

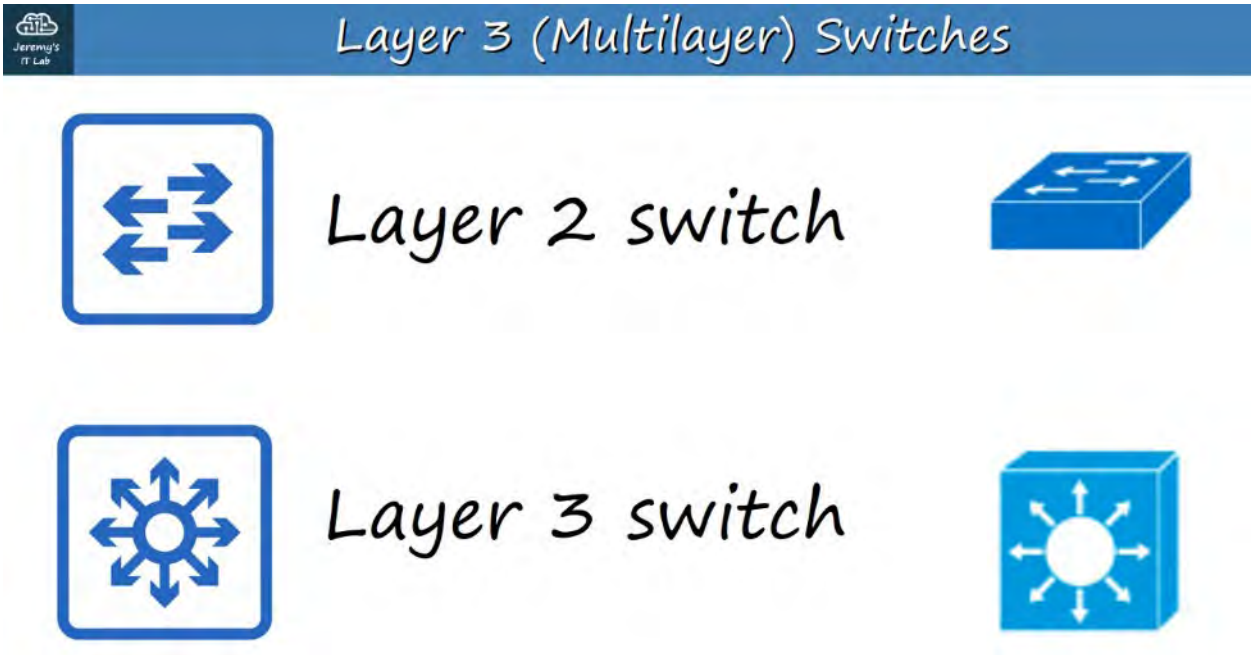
- Configure the IP address for the native VLAN on the router's physical interface (the "**encapsulation dot1q** command is not necessary"

```
R1(config)#no interface g0/0.10
R1(config)#interface g0/0
R1(config-if)#ip address 192.168.1.62 255.255.255.192
R1(config-if)#
```

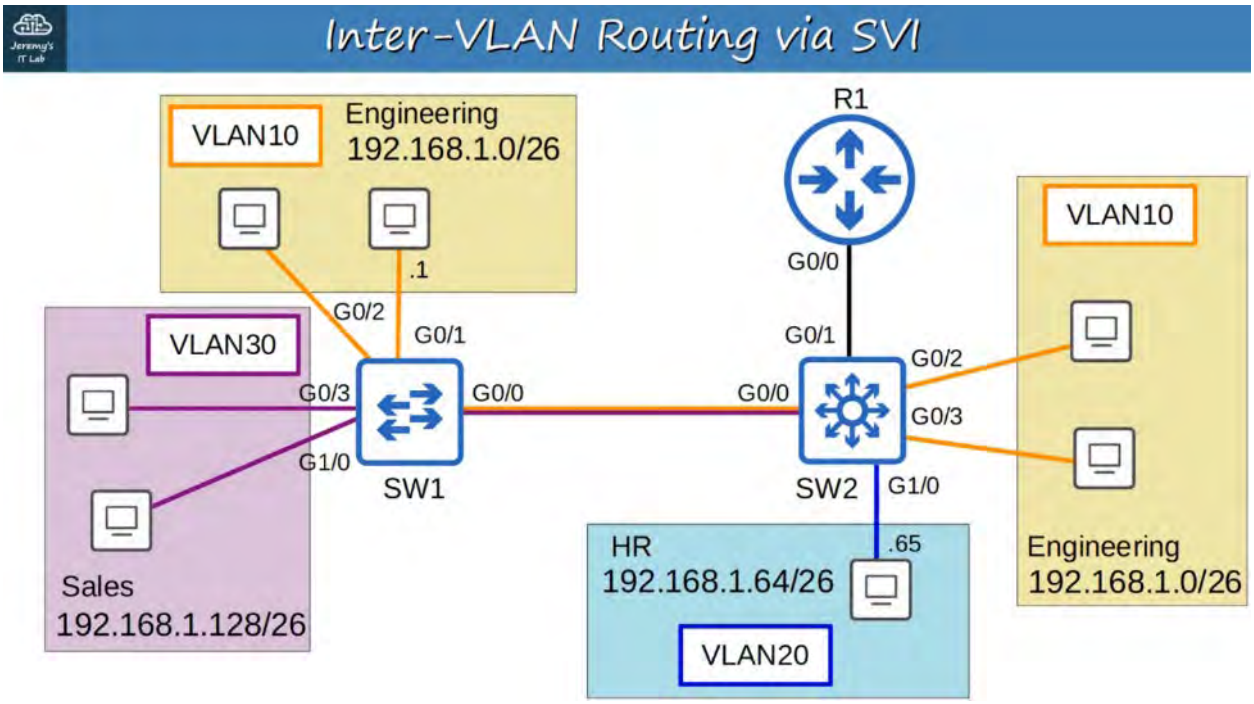
Output of "show running-config" of G0/0 Interface

```
!
interface GigabitEthernet0/0
 ip address 192.168.1.62 255.255.255.192
 duplex auto
 speed auto
 media-type rj45
!
interface GigabitEthernet0/0.20
 encapsulation dot1Q 20
 ip address 192.168.1.126 255.255.255.192
!
interface GigabitEthernet0/0.30
 encapsulation dot1Q 30
 ip address 192.168.1.190 255.255.255.192
!
```

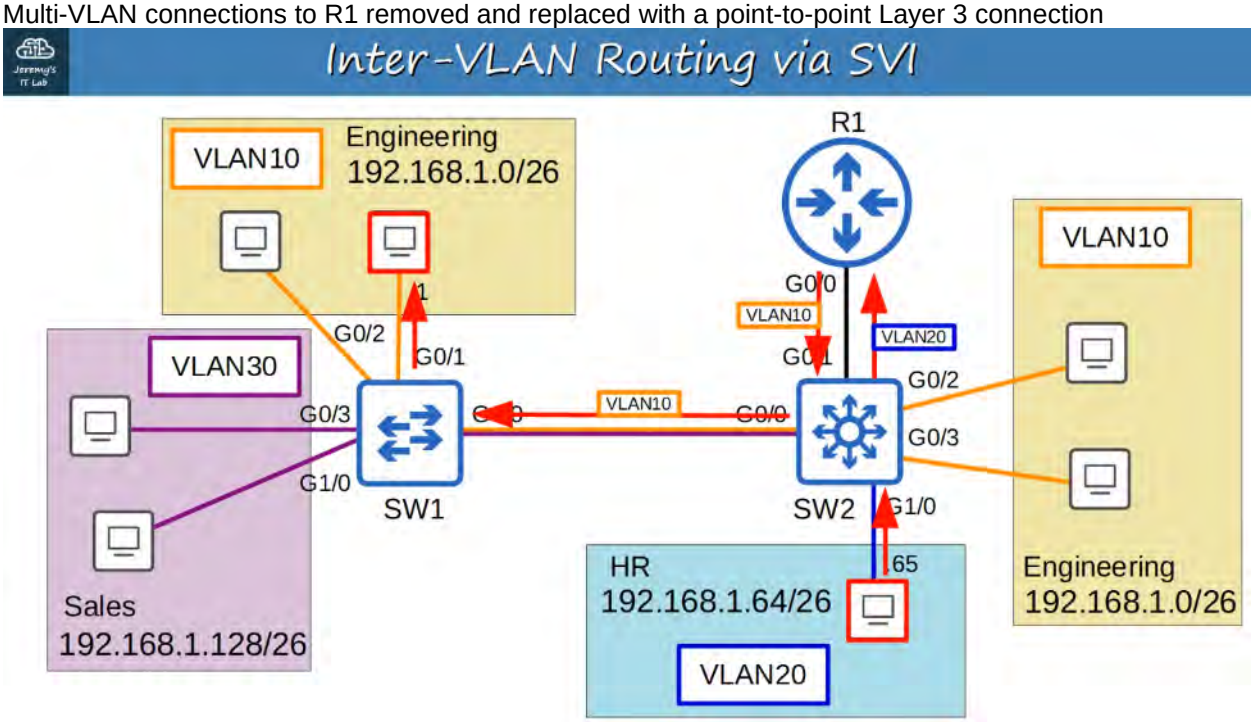
LAYER 3 (MULTILAYER) SWITCHES
ICON APPEARANCE



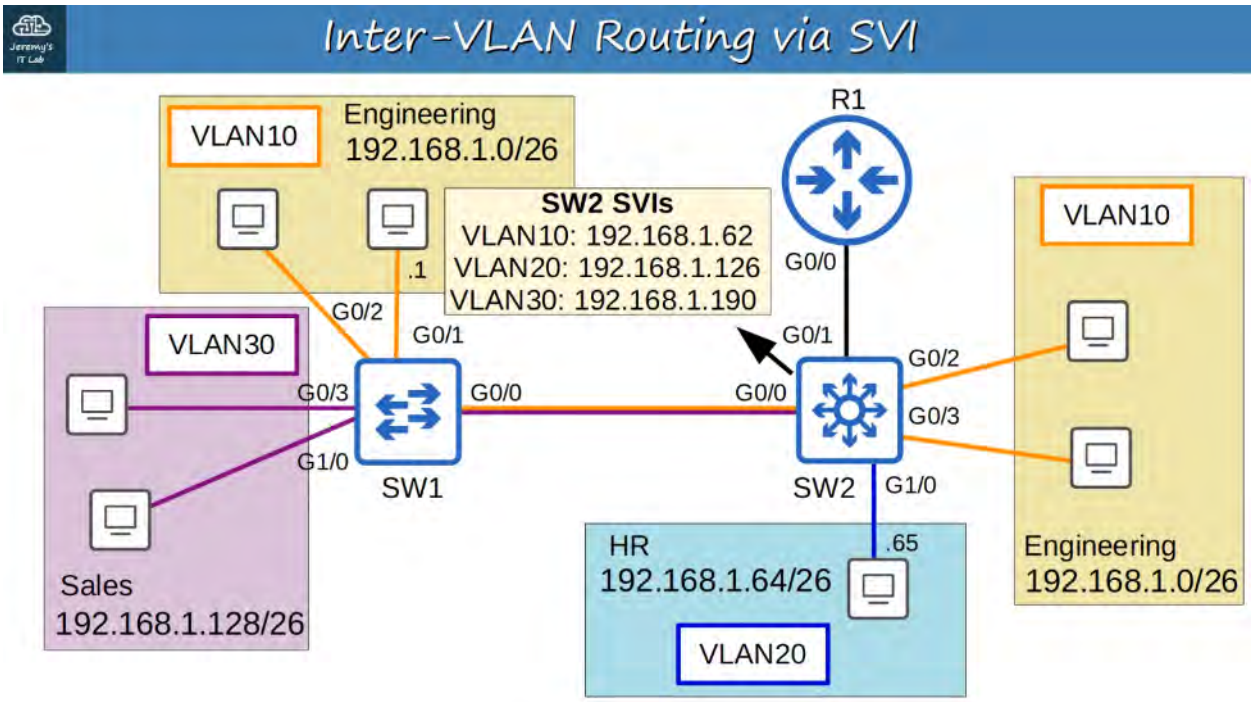
- A MULTILAYER SWITCH is capable of both SWITCHING and ROUTING
- It is LAYER 3 AWARE
- You can assign IP Addresses to its L3 Virtual Interface, like a router
- You can create Virtual Interfaces for each VLAN, and assign IP addresses to those interfaces
- You can configure routes on it, just like a ROUTER
- It can be used for inter-VLAN routing

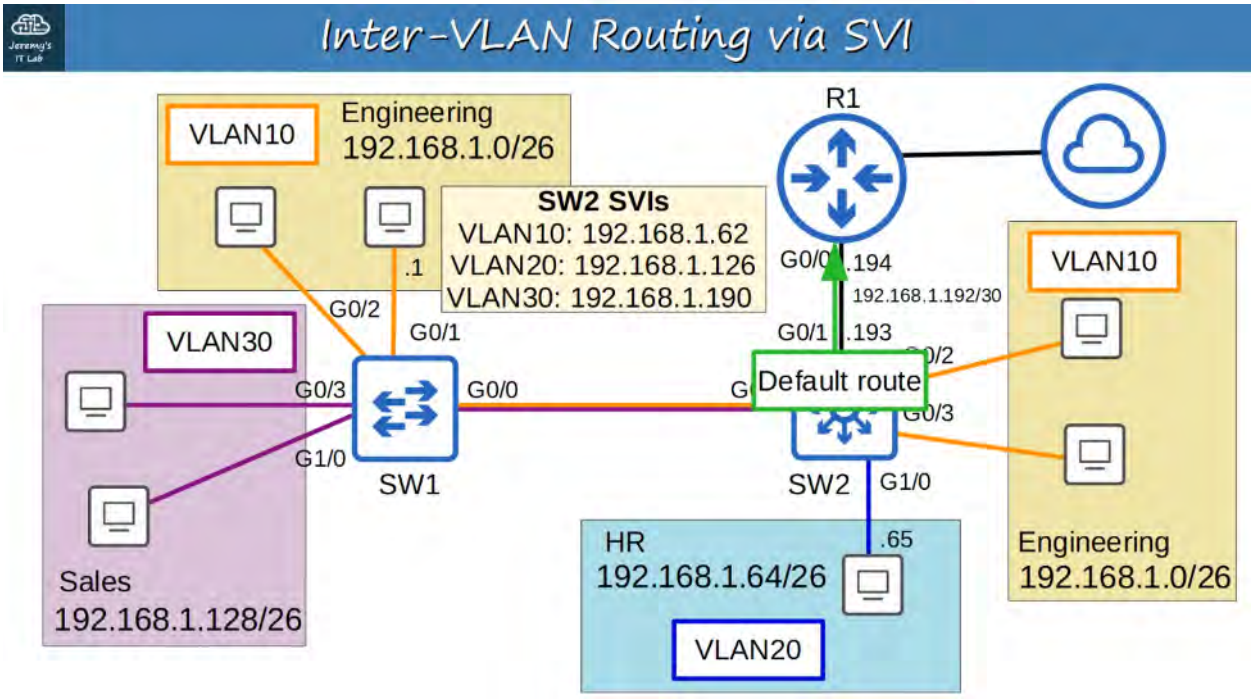


SW2 Replaced with a Layer 3 Switch
Multi-VLAN connections to R1 removed and replaced with a point-to-point Layer 3 connection



- SVIs (Switch Virtual Interfaces) are the virtual interfaces you can assign IP addresses to in a MULTILAYER SWITCH.
- Configure each HOST to use the SVI (NOT the ROUTER R1) as their Gateway Address
- To send traffic to different SUBNETS / VLANS, the PCs will send traffic to the SWITCH, and the SWITCH will route the traffic.





Clearing R1 configuration to set to work with the Layer 3 Point-to-Point connection

Inter-VLAN Routing via SVI

```
R1(config)#no interface g0/0.10
R1(config)#no interface g0/0.20
R1(config)#no interface g0/0.30
R1(config)#default interface g0/0
Interface GigabitEthernet0/0 set to default configuration
R1(config)#do show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         unassigned      YES NVRAM  up                    up
GigabitEthernet0/0.10      unassigned      YES manual deleted               down
GigabitEthernet0/0.20      unassigned      YES manual deleted               down
GigabitEthernet0/0.30      unassigned      YES manual deleted               down
GigabitEthernet0/1         unassigned      YES NVRAM  administratively down down
GigabitEthernet0/2         unassigned      YES NVRAM  administratively down down
GigabitEthernet0/3         unassigned      YES NVRAM  administratively down down
R1(config)#
```

#no interface : removes the VLAN interface
#default interface g0/0 : resets the g0/0 interface to it's default settings
Then configure the default R1 G0/0 interface's to IP address : 192.168.1.194 (as per the network diagram)
Configuration of SW2 to use SVI and the Layer 3 Point-to-Point connection with R1

Inter-VLAN Routing via SVI

```
SW2(config)#default interface g0/1
Interface GigabitEthernet0/1 set to default configuration
SW2(config)#ip routing
SW2(config)#interface g0/1
SW2(config-if)#no switchport
SW2(config-if)#ip address 192.168.1.193 255.255.255.252
SW2(config-if)#do show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES unset  up                    up
GigabitEthernet0/2     unassigned      YES unset  up                    up
GigabitEthernet0/3     unassigned      YES unset  up                    up
GigabitEthernet0/1     192.168.1.193   YES manual up                    up
GigabitEthernet1/0     unassigned      YES unset  up                    up
GigabitEthernet1/1     unassigned      YES unset  up                    up
GigabitEthernet1/2     unassigned      YES unset  up                    up
GigabitEthernet1/3     unassigned      YES unset  up                    up
GigabitEthernet2/0     unassigned      YES unset  up                    up
GigabitEthernet2/1     unassigned      YES unset  up                    up
GigabitEthernet2/2     unassigned      YES unset  up                    up
GigabitEthernet2/3     unassigned      YES unset  up                    up
```

"default interface " : resets settings on specified interface to defaults
"ip routing" : **IMPORTANT** command to enable Layer 3 routing on the SWITCH
"no switchport" : configures the interface from a Layer 2 Switchport to a Layer 3 "routed port"
The sets the Default Route to R1 (192.168.1.194) so that all traffic leaving the network gets routed through R1's Gateway of Last Resort (aka The Default Gateway)

Inter-VLAN Routing via SVI

```
SW2(config-if)#exit
SW2(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.194
SW2(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
       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

Gateway of last resort is 192.168.1.194 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.168.1.194
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.192/30 is directly connected, GigabitEthernet0/1
L        192.168.1.193/32 is directly connected, GigabitEthernet0/1
SW2(config)#
```

Inter-VLAN Routing via SVI

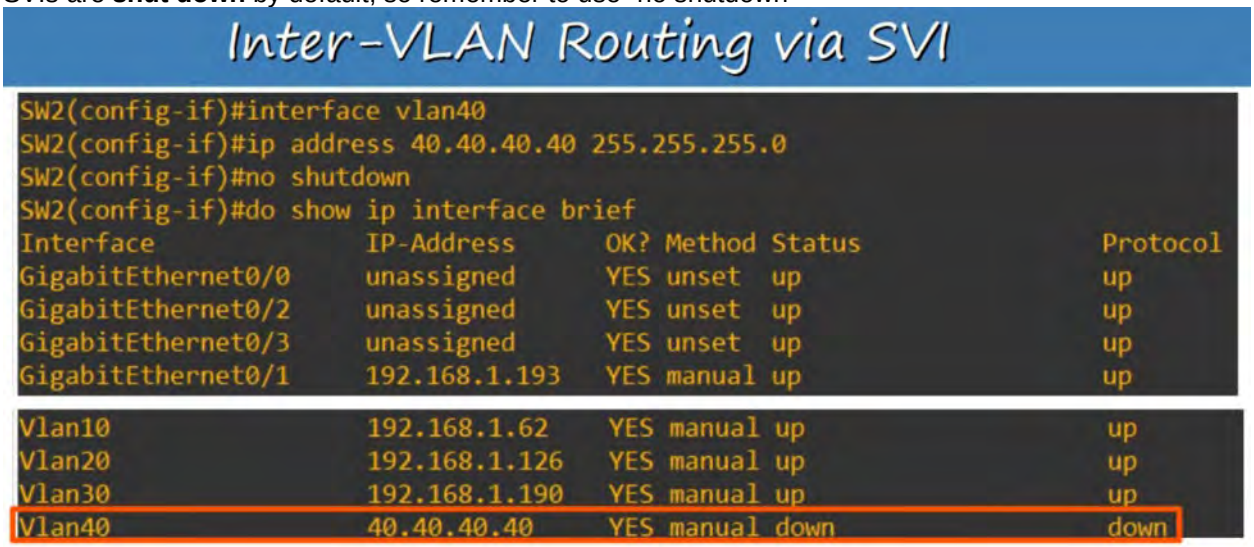
```
SW2#show interfaces status

Port      Name               Status       Vlan       Duplex  Speed Type
Gi0/0                        connected    trunk        auto   auto unknown
Gi0/2                        connected    10           auto   auto unknown
Gi0/3                        connected    10           auto   auto unknown
Gi0/1                        connected    routed       auto   auto unknown
Gi1/0                        connected    20           auto   auto unknown
Gi1/1                        connected    1            auto   auto unknown
Gi1/2                        connected    1            auto   auto unknown
Gi1/3                        connected    1            auto   auto unknown
Gi2/0                        connected    1            auto   auto unknown
Gi2/1                        connected    1            auto   auto unknown
Gi2/2                        connected    1            auto   auto unknown
Gi2/3                        connected    1            auto   auto unknown
Gi3/0                        connected    1            auto   auto unknown
Gi3/1                        connected    1            auto   auto unknown
Gi3/2                        connected    1            auto   auto unknown
Gi3/3                        connected    1            auto   auto unknown
SW2#
```

SVI CONFIGURATION ON SW2 (Virtual LAYER 3 ROUTING INTERFACES)

Inter-VLAN Routing via SVI

```
SW2(config)#interface vlan10
SW2(config-if)#ip address 192.168.1.62 255.255.255.192
SW2(config-if)#no shutdown
SW2(config-if)#interface vlan20
SW2(config-if)#ip address 192.168.1.126 255.255.255.192
SW2(config-if)#no shutdown
SW2(config-if)#interface vlan30
SW2(config-if)#ip address 192.168.1.190 255.255.255.192
SW2(config-if)#no shutdown
```

SVIs are **shut down** by default, so remember to use “no shutdown”



Creating an unknown SVI (VLAN 40) and the Status/Protocol is “down/down”
What are the conditions for a SVI to be “up/up” ?

- The VLAN must exist on the SWITCH
- The SWITCH must have at least ONE access port in the VLAN in an “up/up” state AND/OR one TRUNK port that allows the VLAN that is in an “up/up” state
- The VLAN must not be shutdown (you can use the “shutdown” command to disable a VLAN)
- The SVI must not be shutdown (SVIs are disabled, by default)

Inter-VLAN Routing via SVI

```
SW2(config-if)#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
       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

Gateway of last resort is 192.168.1.194 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 192.168.1.194
      192.168.1.0/24 is variably subnetted, 8 subnets, 3 masks
C        192.168.1.0/26 is directly connected, Vlan10
L        192.168.1.62/32 is directly connected, Vlan10
C        192.168.1.64/26 is directly connected, Vlan20
L        192.168.1.126/32 is directly connected, Vlan20
C        192.168.1.128/26 is directly connected, Vlan30
L        192.168.1.190/32 is directly connected, Vlan30
C        192.168.1.192/30 is directly connected, GigabitEthernet0/1
L        192.168.1.193/32 is directly connected, GigabitEthernet0/1
SW2(config-if)#
```

The VLAN trunk has been successfully replaced by an Layer 3 SWITCH SVI.
All hosts should be able to connect with each other (tested with “ping”) as well as reach the external internet (via the Cloud symbol attached to R1)

19. DTP / VTP (Not in Syllabus)

DTP (Dynamic Trunking Protocol)

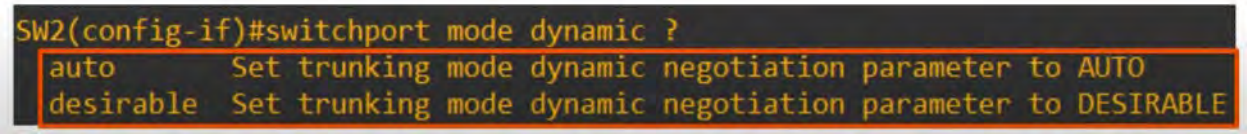
- Protocol that allows SWITCHES to negotiate the status of their SWITCHPORTS, without manual configuration, to be:
  - ACCESS PORTS
  - TRUNK PORTS
- DTP is ENABLED by default on all Cisco SWITCH interfaces

We've been manually configuring SWITCHPORTS using :

- "switchport mode access"
- "switchport mode trunk"

💡 'show interfaces <interface-id> switchport' will show you a switchport's settings.

For security purposes, **manual configuration** is recommended. DTP should be disabled on ALL SWITCHPORTS



DYNAMIC DESIRABLE:

- This MODE will actively try to form a TRUNK with other Cisco SWITCHES.
- Will form a TRUNK if connected to another SWITCHPORT in the following modes:
  - "switchport mode trunk"
  - "switchport mode dynamic desirable"
  - "switchport mode dynamic auto"

HOWEVER … if the other interface is set to "static access" (ACCESS mode), it will NOT form a TRUNK, it will be an ACCESS PORT

DYNAMIC AUTO:

- This MODE will NOT actively try to form a TRUNK with other Cisco SWITCHES
- Will form a TRUNK if connected SWTICH is actively trying to form a TRUNK.
- It will form a TRUNK with a SWITCHPORT in the following modes:
  - "switchport mode trunk"
  - "switchport mode dynamic desirable"

TRUNK to ACCESS connection will operate in a **Mismatched Mode**.

This configuration does NOT work and SHOULD result in an error. Traffic will NOT work.

TABLE SHOWING THE DIFFERENT MODES AND COMPATIBILITY IN FORMING A TRUNK

Administrative Mode	Trunk	Dynamic Desirable	Access	Dynamic Auto
Trunk	Trunk	Trunk	X	Trunk
Dynamic Desirable	Trunk	Trunk	Access	Trunk
Access	X	Access	Access	Access
Dynamic Auto	Trunk	Trunk	Access	Access

DTP will NOT form a TRUNK with:
a ROUTER
a PC
etcetera …
The SWITCHPORT will be in ACCESS Mode only!
OLD SWITCHES:

- “switchport mode dynamic desirable” = Default administrative mode.

NEWER SWITCHES:

- “switchport mode dynamic auto” = Default administrative mode.

HOW TO DISABLE DTP NEGOTIATION ON AN INTERFACE:

- “switchport nonegotiate”
- “switchport mode access”

It is a security recommendation to disable DTP on all SWITCHPORTS and manually configure them as ACCESS or TRUNK ports.

ENCAPSULATION:
SWITCHES that support both:

- 802.1Q
- ISL

TRUNK encapsulation can use DTP to negotiate the encapsulation they will use.

- Negotiation is Enabled by default

💡 'switchport trunk encapsulation negotiate'

- ISL is favored over 802.1Q
  - If BOTH SWITCHES support ISL, ISL will be selected.
- DTP frames are sent in:
  - VLAN1 when using ISL
  - Native VLAN when using 802.1Q (the default native VLAN is VLAN1, however)

VTP (VLAN Trunking Protocol)
In Privileged EXEC mode:

💡 #show vtp status

- Protocol for configuring VLANs on a Central SWITCH
  - A SERVER that other SWITCHES synch. to (auto configuring by connection)
- Other switches (VTP CLIENTS) will synchronize their VLAN database to the SERVER
- Designed for large networks with many VLANs (reduces manual configuration)
- RARELY used. Recommended you DO NOT USE it
- There are THREE VTP Versions :
  - v1
    - Does NOT supports Extended VLAN Range 1006-4094
  - v2
    - Does NOT supports Extended VLAN Range 1006-4094
    - Supports Token Ring VLANs ; otherwise similar to V1
  - v3
    - Supports Extended VLAN Range 1006-4094
    - CLIENTS store VLAN dBase in NVRAM
- There are **THREE VTP modes**:
  - SERVER
  - CLIENT
  - TRANSPARENT
- Cisco SWITCHES operate in VTP SERVER mode, by default

Server	Client	Transparent
creates/modifies/deletes VLANs	synchronizes VTP information	creates/modifies/deletes VLANs
synchronizes VTP information	originates VTP advertisements	forwards VTP advertisements
originates VTP advertisements	forwards VTP advertisements	stores VLAN information in NVRAM
forwards VTP advertisements		
stores VLAN information in NVRAM		

VTP SERVERS:

- Can ADD / MODIFY / DELETE VLANs
- Store the VLAN dBase in NVRAM
- Increase Revision Number every time VLAN is Added / Modified / Deleted
- Advertises **Latest Version** of VLAN dBase on TRUNK interfaces.
- VTP CLIENTS synchronize their VLAN dBase to it
- **VTP SERVERS also function as VTP CLIENTS**
    - **THEREFORE, a VTP SERVER will synchronize to another VTP SERVER with a higher Revision Number**

🚨 One danger of VTP: Connecting an old SWITCH with higher Revision Number to network (and if the VTP Domain Name matches), all SWITCHES in Domain will synchronize their VLAN dBase to SWITCH

VTP CLIENTS:

💡 (config)# vtp mode client

- Cannot Add / Modify / Delete VLANs
- Does NOT store the VLAN database in NVRAM
    - **VTP v3 CLIENTS DO**
- Will synchronize their VLAN dBase to the SERVER with the highest version number in their VTP Domain
- Advertise their VLAN dBase and forward VTP Advertisements to other CLIENTS over TRUNK Ports

VTP TRANSPARENT MODE:

💡 (config)# vtp mode transparent

- Does NOT participate in VTP Domain (does NOT sync VLAN database)
- Maintains own VLAN dBase in NVRAM.
- Can Add / Modify / Delete VLANs
- Won't Advertise to other SWITCHES
- Will forward VTP advertisements to SWITCHES in the same Domain as it.

VTP DOMAINS

If a SWITCH with no VTP Domain (Domain NULL) receives a VTP advertisement with a VTP Domain name, it will automatically join that VTP Domain

If a SWITCH receives a VTP advertisement in the same VTP domain with a higher revision number, it will update it's VLAN database to match

---

REVISION NUMBERS:

There are TWO ways to RESET a REVISION NUMBER to 0:

- Change VTP Domain to an unused Domain
- Change VTP mode to TRANSPARENT

VTP VERSION NUMBER

💡 (config)#vtp version <version number>

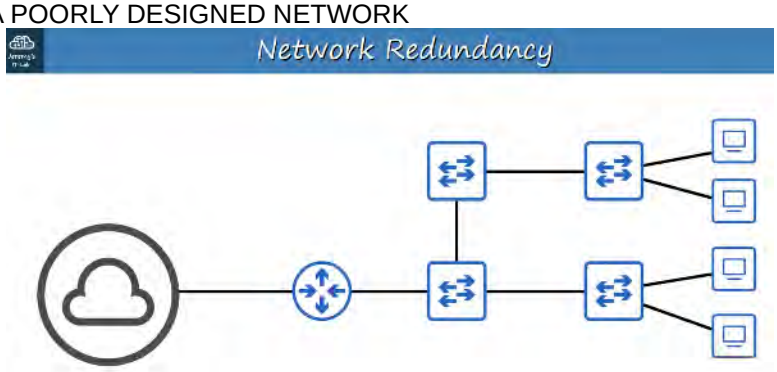
Changing the Version # will force sync/update all connected SWITCHES to the latest Version #

20. SPANNING TREE PROTOCOL (STP) : PART 1

REDUNDANCY IN NETWORKS

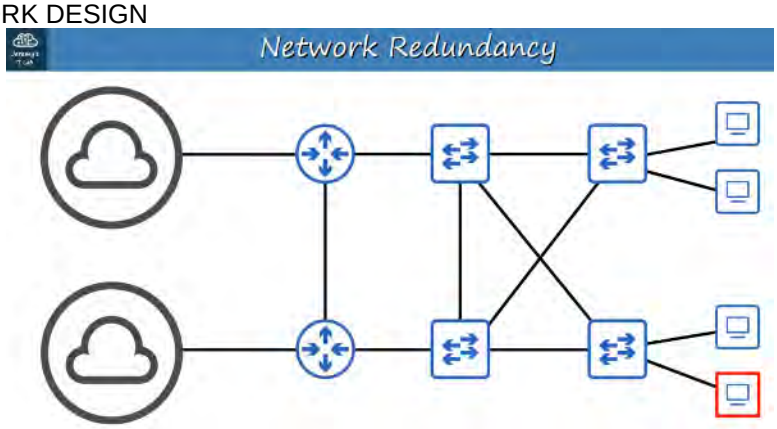
- Essential in network design
- Modern networks are expected to run 24/7/265; even a short downtime can be disastrous for business.
- If one network component fails, you must ensure that other components will take over with little or no downtime.
- As much as possible, you must implement REDUNDANCY at every possible point in the network

AN EXAMPLE OF A POORLY DESIGNED NETWORK



NOTE the many single-point failures that could occur (single connections)

A BETTER NETWORK DESIGN



UNFORTUNATELY :

- Most PCS only have a single network interface card (NIC), so they can only be plugged into a single SWITCH. However, important SERVERS typically have multiple NICs, so they can be plugged into multiple SWITCHES for redundancy!

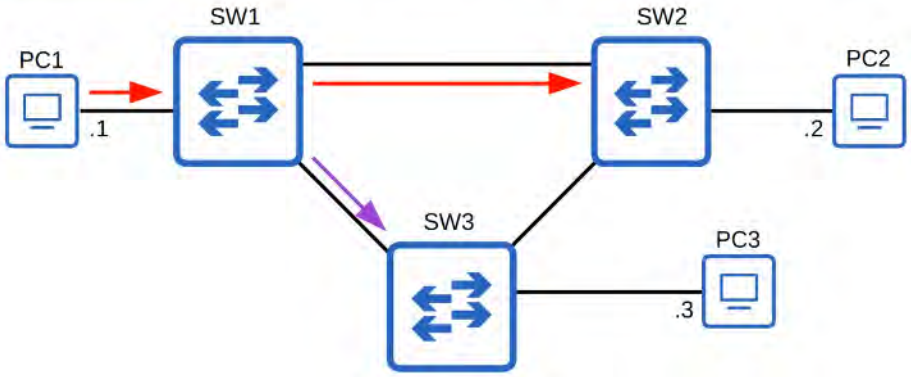
So HOW can all this redundancy be a BAD thing?

BROADCAST STORMS

Broadcast Storms

ARP Request
Dst: FFFF.FFFF.FFFF

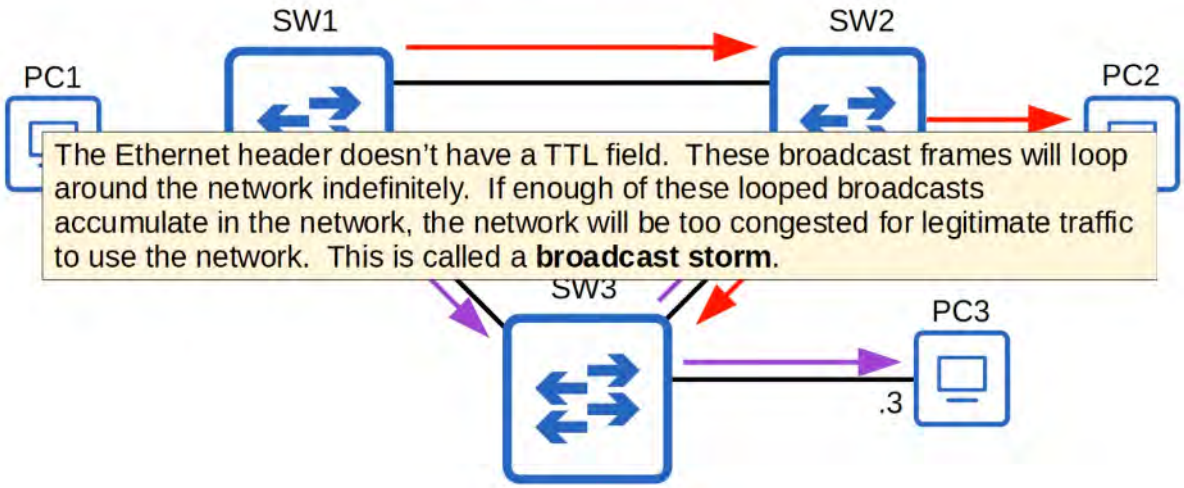
10.0.0.0/24



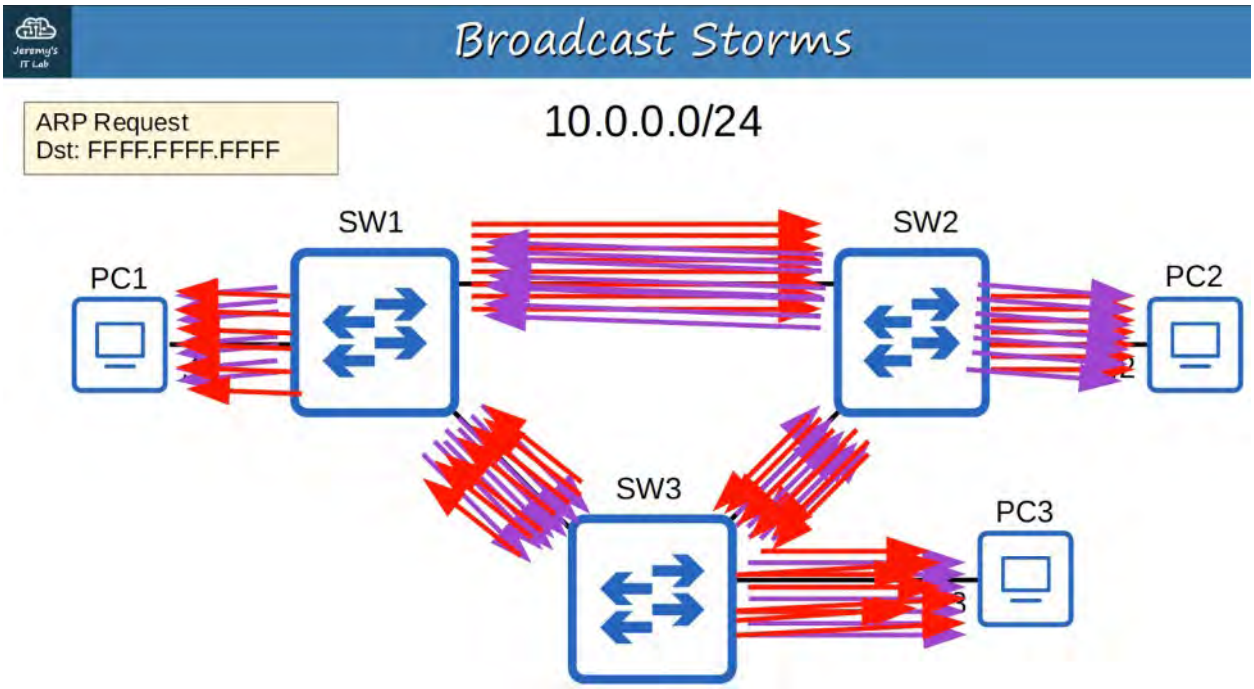
Broadcast Storms

ARP Request
Dst: FFFF.FFFF.FFFF

10.0.0.0/24



The Ethernet header doesn't have a TTL field. These broadcast frames will loop around the network indefinitely. If enough of these looped broadcasts accumulate in the network, the network will be too congested for legitimate traffic to use the network. This is called a **broadcast storm**.

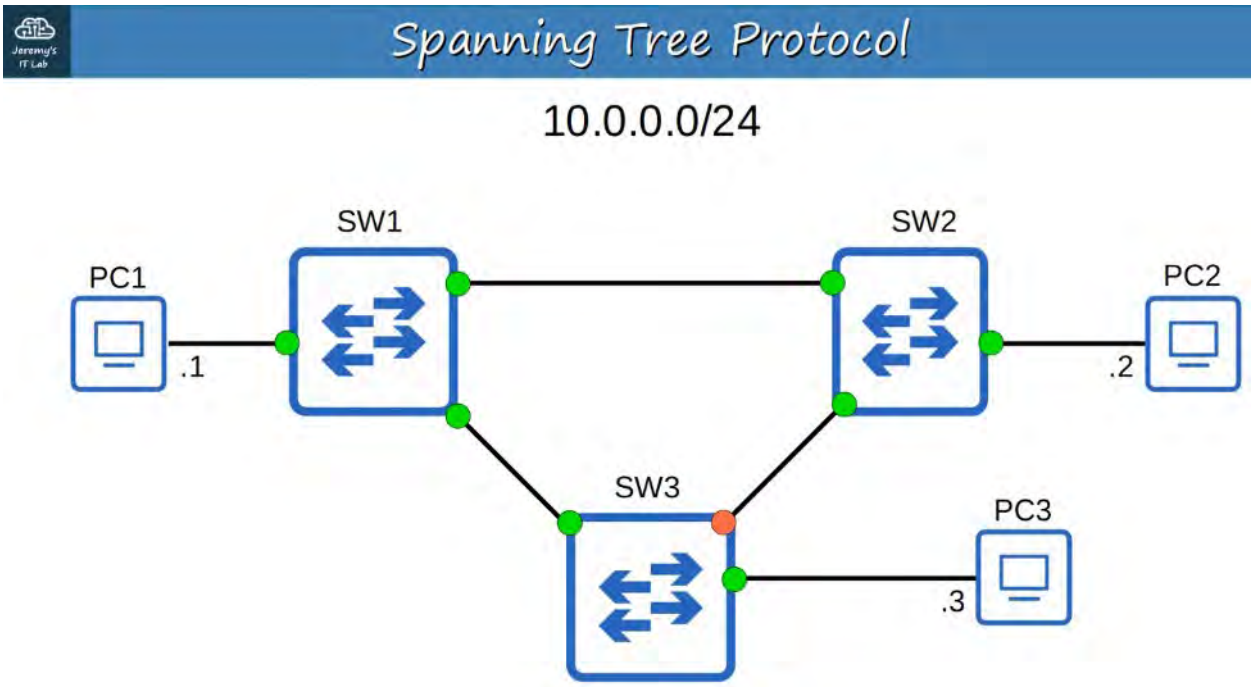


FLOODED WITH ARP REQUESTS (Red = Clockwise Loops // Purple = Counter-Clockwise Loops)
Network Congestion isn't the only problem.
Each time a FRAME arrives on a SWITCHPORT, the SWITCH uses the SOURCE MAC ADDRESS field to "learn" the MAC ADDRESS and update it's MAC ADDRESS TABLE.
When frames with the same SOURCE MAC ADDRESS repeatedly arrive on different interfaces, the SWITCH is continuously updating the interface in it's MAC ADDRESS TABLE.
This is called MAC ADDRESS FLAPPING
So how we design a network, with redundant paths, that doesn't result in LAYER 2 LOOPS.
SPANNING TREE PROTOCOL is one solution

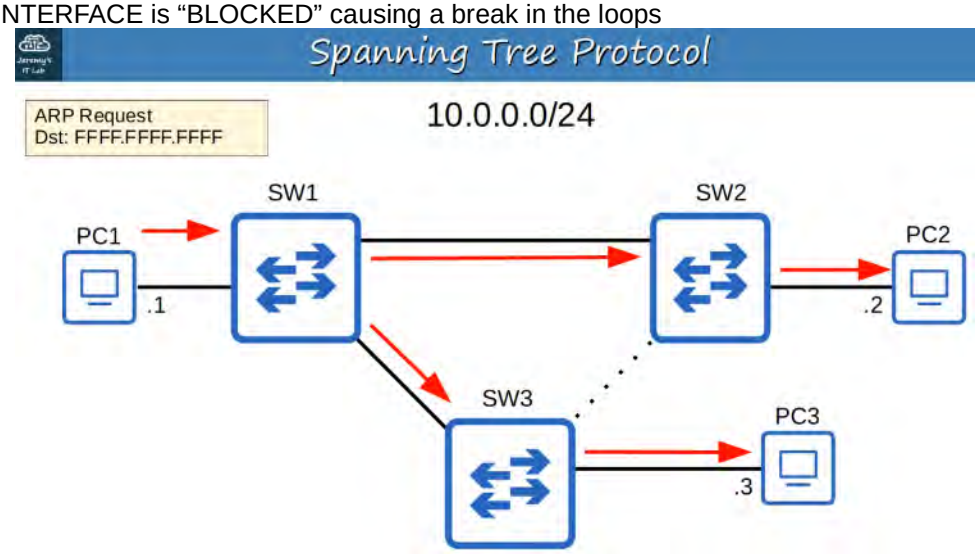
STP (SPANNING TREE PROTOCOL) : 802.1D

- "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 BLOCKING state only send or receive STP messages (called BPDUs = Bridge Protocol Data Units)

💡 SPANNING TREE PROTOCOL still uses the term "BRIDGE". However, when use the term "BRIDGE", we really mean "SWITCH". BRIDGES are not used in modern networks.



ORANGE INTERFACE is “BLOCKED” causing a break in the loops



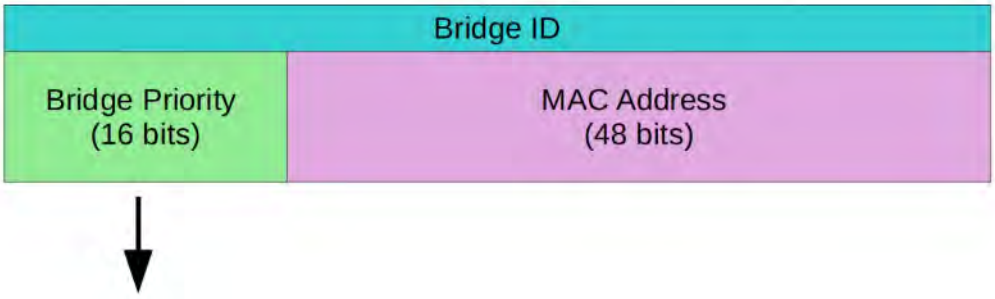
If changes occur in the connections, the traffic will adjust the topology.

- By selecting WHICH ports are FORWARDING and which ports are BLOCKING, STP creates a single path TO / FROM each point in the NETWORK. This prevents LAYER 2 Loops.
- There is a set process that STP uses to determine which ports should be FORWARDING and which should be BLOCKING
- STP-enabled SWITCHES send / receive “Hello BPDUs” out of all INTERFACES
    - The default timer is : ONCE every TWO seconds per INTERFACE!
- If a SWITCH receives a “Hello BPDU” on an INTERFACE, it knows that INTERFACE is connected to another SWITCH (ROUTERS, PCs, etc. do NOT use STP so do not send “Hello BPDUs”)

WHAT ARE BPDUs USED FOR?

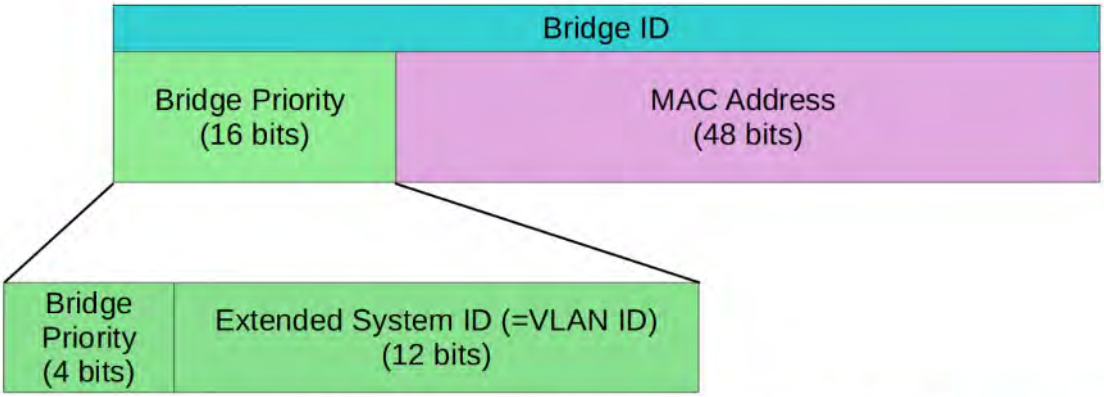
- SWITCHES use one field in the STP BPDU, the BRIDGE ID field, to elect a ROOT BRIDGE for the NETWORK
- The SWITCH with the lowest BRIDGE ID becomes the ROOT BRIDGE

- 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



The default bridge priority is 32768 on all switches, so by default the MAC address is used as the tie-breaker (lowest MAC address becomes the root bridge).

The Bridge Priority is compared first. If they tie, the MAC address is then compared



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.

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

In the default VLAN of 1, the default bridge priority is actually **32769** (32768 + 1).

If you want to change the switch's bridge priority (without changing VLAN numbers), what is the minimum unit of increase/decrease?

To REDUCE the BRIDGE PRIORITY, we can only change it in units of 4096 !

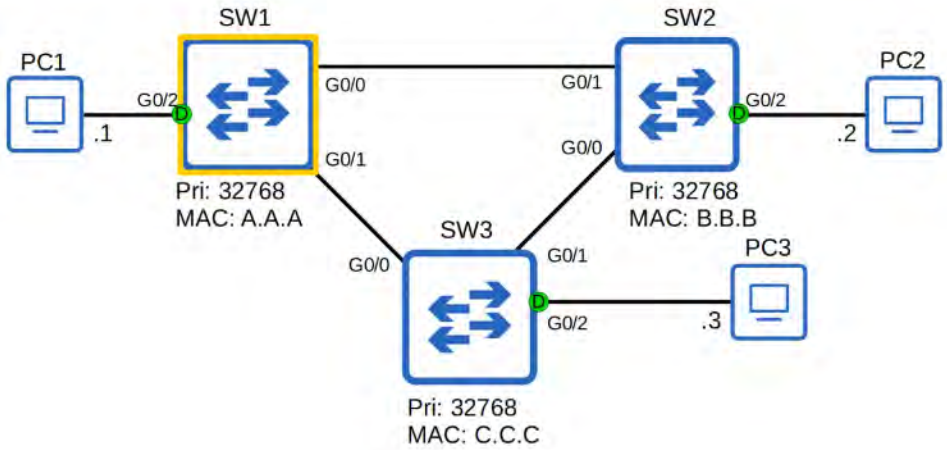
Bridge Priority				Extended System ID (VLAN ID)											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1

= 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.

In THIS TOPOLOGY, SW1 becomes the ROOT BRIDGE due to it's MAC ADDRESS being LOWEST (Hex "A" = 10)



ALL INTERFACES on the ROOT BRIDGE are DESIGNATED PORTS.
DESIGNATED PORTS ARE IN A FORWARDING STATE!
ROOT BRIDGE

- 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 STEPS

1. One SWITCH is elected as ROOT BRIDGE. All PORTS on the ROOT BRIDGE are DESIGNATED PORTS (FORWARDING STATE)

- ROOT BRIDGE selection order:
  - 1. Lowest BRIDGE ID
  - 2. Lowest MAC Address (in case of Bridge ID tie)

2. Each remaining SWITCH will select ONE of its INTERFACES to be it's ROOT PORT (FORWARDING STATE). PORTS across from the ROOT PORT are always DESIGNATED PORTS.

- ROOT PORT selection order:
    - 1. LOWEST ROOT COST (see STP COST CHART)
    - 2. LOWEST NEIGHBOUR BRIDGE ID
    - 3. LOWEST NEIGHBOUR PORT ID

3. Each remaining COLLISION DOMAIN will select ONE INTERFACE to be a DESIGNATION PORT (FORWARDING STATE). The other PORT in the COLLISION DOMAIN will NON-DESIGNATED (BLOCKING)

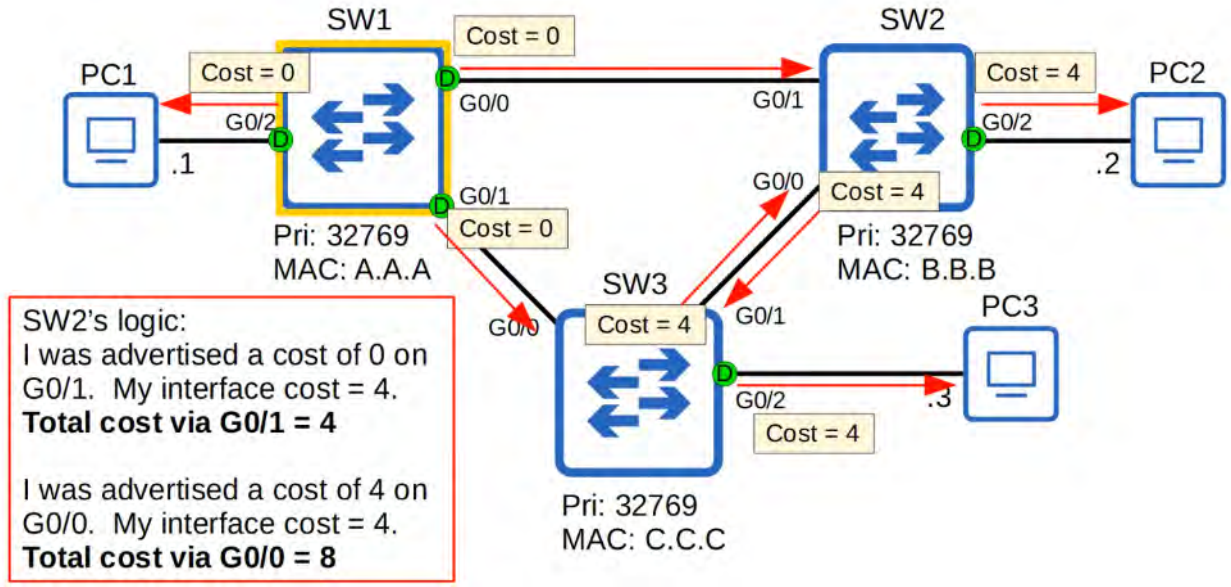
- DESIGNATED PORT SELECTION:
    - 1. INTERFACE on SWITCH with LOWEST ROOT COST
    - 2. INTERFACE on SWITCH with LOWEST BRIDGE ID

STP COST CHART

💡 Only OUTGOING INTERFACES toward the ROOT BRIDGE have a STP COST; not RECEIVING INTERFACES. Add up all the OUTGOING PORT costs until you reach the ROOT BRIDGE

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

SW1 is the ROOT BRIDGE so has a STP COST of 0 on ALL INTERFACES



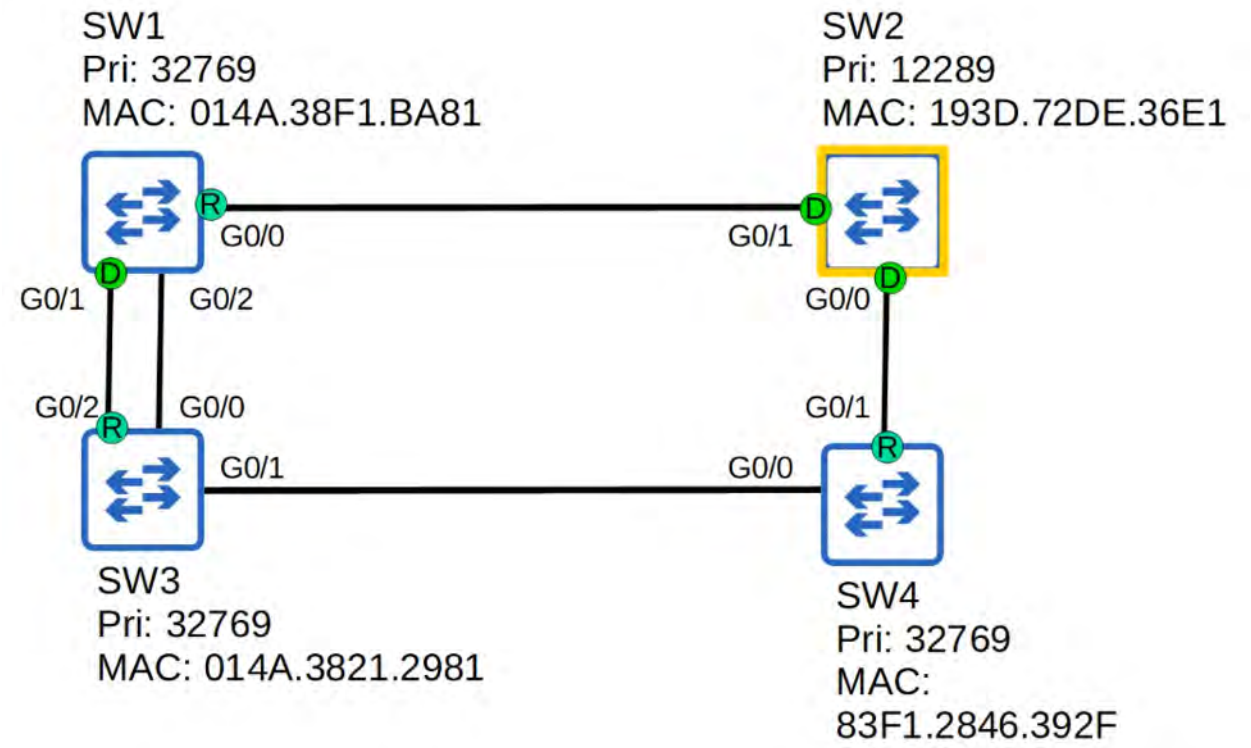
The PORTS connected to another SWITCH's ROOT PORT MUST be DESIGNATED (D).
Because the ROOT PORT Is the SWITCH's path to the ROOT BRIDGE, another SWITCH must not block it.
STP PORT ID (in case of a tie-breaker)

```
SW1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     aaaa.aaaa.aaaa
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
```

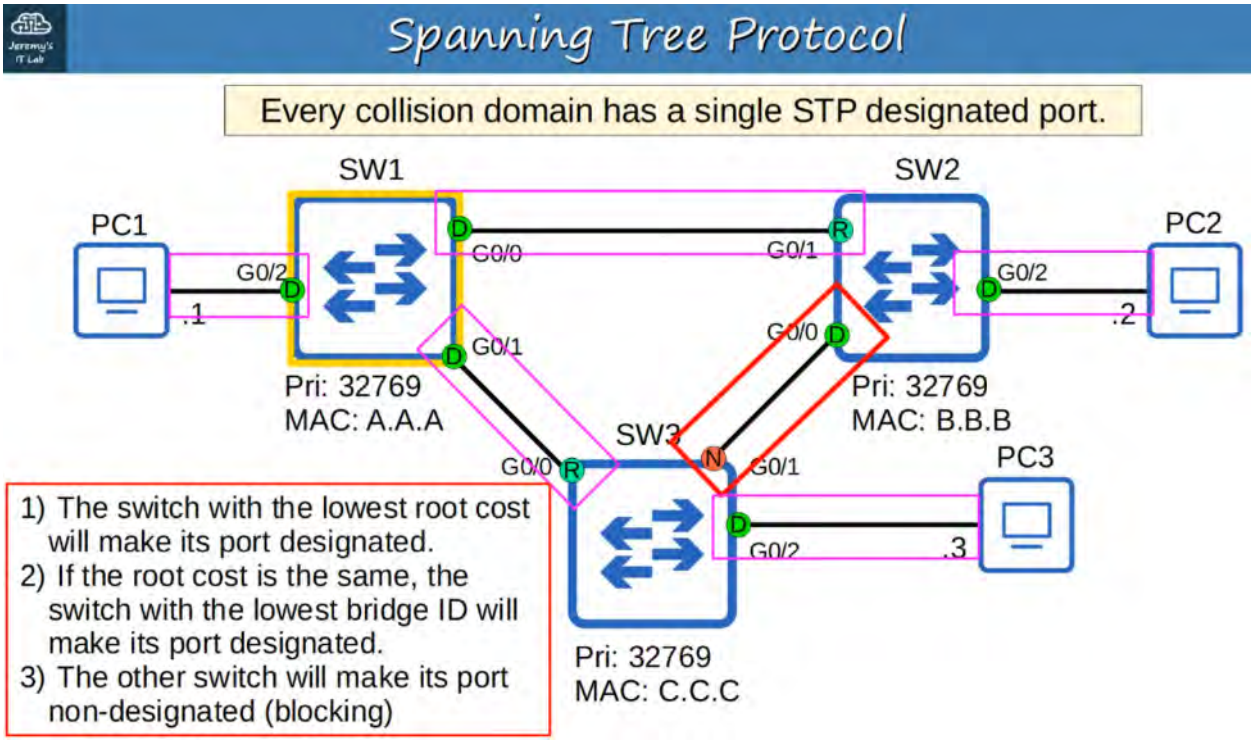
STP Port ID = port priority (default 128) + port number

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi0/0	Desg	FWD	4	128.1	Shr
Gi0/1	Desg	FWD	4	128.2	Shr
Gi0/2	Desg	FWD	4	128.3	Shr
Gi0/3	Desg	FWD	4	128.4	Shr
Gi1/0	Desg	FWD	4	128.5	Shr
Gi1/1	Desg	FWD	4	128.6	Shr
Gi1/2	Desg	FWD	4	128.7	Shr
Gi1/3	Desg	FWD	4	128.8	Shr

NEIGHBOUR SWITCH PORT ID (in case of a tie-breaker)
(D) = DESIGNATED PORT
(R) = ROOT PORT



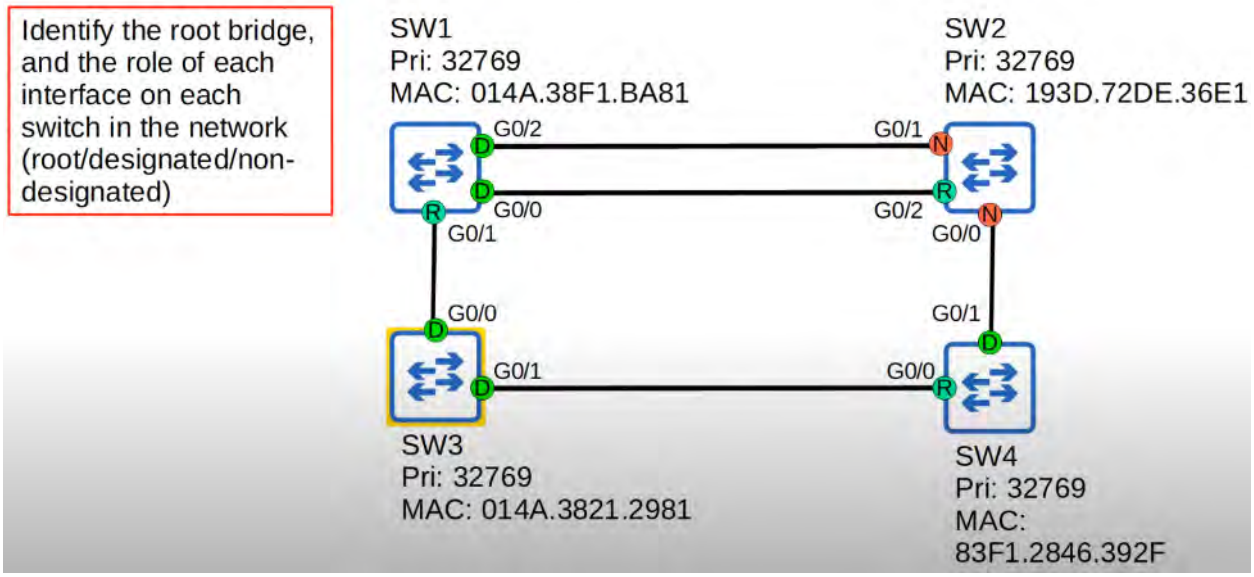
HOW TO DETERMINE WHICH PORT WILL BE BLOCKED TO PREVENT LAYER 2 LOOPS



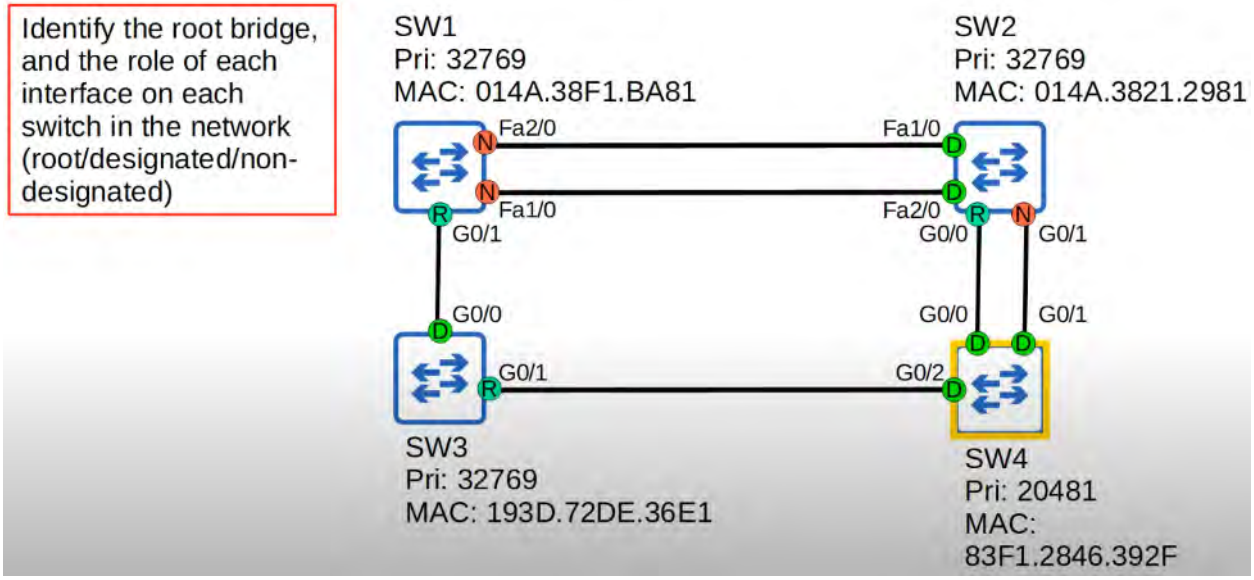
QUIZ

Identify the ROOT BRIDGE and the ROLE of EACH INTERFACE on the NETWORK (ROOT / DESIGNATED / NON-DESIGNATED)

#1



ALL SWITCHES have the same PRIORITY NUMBER (32769)
Tie-breaker goes to the LOWEST MAC ADDRESS
SW3 has the LOWEST so it's the ROOT BRIDGE and ALL it's INTERFACES become DESIGNATED
Connections from SW1 (G0/1) and S4 (G0/0) to SW3 become ROOT INTERFACES
Because SW2 has TWO connections to SW1, both of SW1's INCOMING interfaces become DESIGNATED.
SW2 G0/2 INTERFACE becomes a ROOT INTERFACE because the G0/0 INTERFACE of SW1 is LOWER than it's G0/2 INTERFACE
The remaining interfaces on SW2 become NON-DESIGNATED because it has the HIGHEST ROOT COST (12 = 4x 1 GB connection). INTERFACES they are attached to on other SWITCHES become DESIGNATED
#2



SW4 has the LOWEST Priority Number so it is designated ROOT BRIDGE
All of SW4 INTERFACES become DESIGNATED
SW2 G0/0 becomes ROOT PORT because SW4 G0/0 connection is a LOWER NUMBER than G0/1.
SW3 G0/1 becomes ROOT PORT
SW1 G0/1 becomes ROOT PORT because G0/1 cost is LESS than Fa1/0 and 2/0
EACH remaining PORT will be either DESIGNATED or NON-DESIGNATED

SW1 Fa1/0 and 2/0 become NON-DESIGNATED since they have a HIGHER STP COST (38) than SW2 outbound ports (8) making SW2 Fa1/0 and 2/0 DESIGNATED
SW2 remaining connection, G0/1, NON-DESIGNATED

21. SPANNING TREE PROTOCOL (STP) : PART 2

STP 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 network topology.

1) BLOCKING / STABLE

- NON-DESIGNATED PORTS are in a BLOCKING state
- Interfaces in a BLOCKING state are effectively disabled to prevent loops
- Interfaces in a BLOCKING state do NOT Send/Receive regular network traffic
- Interfaces in a BLOCKING state do NOT forward STP BPDUs
- Interfaces in a BLOCKING state do NOT learn MAC ADDRESSES

2) LISTENING / TRANSITIONAL

- After the BLOCKING state, interfaces with the DESIGNATED or ROOT role enter the LISTENING state
- ONLY DESIGNATED or ROOT PORTS enter the LISTENING state (NON-DESIGNATED PORTS are ALWAYS BLOCKING)
- The LISTENING state is 15 seconds long by Default. This is determined by the FORWARD DELAY TIMER
- Interfaces in a LISTENING state do NOT Send / Receive regular network traffic
- Interfaces in a LISTENING state ONLY Forward/Receive STP BPDUs
- Interfaces in a LISTENING state does NOT learn MAC ADDRESSES from regular traffic that arrives on the interface

3) LEARNING / TRANSITIONAL

- After the LISTENING state, a DESIGNATED or ROOT port will enter the LEARNING state
- The LEARNING state is 15 seconds long by Default. This is determined by the FORWARD DELAY TIMER (same one used for both LISTENING and LEARNING states)
- Interfaces in a LEARNING state do NOT Send / Receive regular network traffic
- Interfaces in a LEARNING state ONLY Sends/Receives STP BPDUs
- Interfaces in a LEARNING state **learns** MAC ADDRESSES from regular traffic that arrives on the interface

4. FORWARDING / STABLE

- ROOT and DESIGNATED PORTS are in a FORWARDING state
- A PORT in the FORWARDING state operate as NORMAL
- A PORT in the FORWARDING state Sends/Receives regular network traffic
- A PORT in the FORWARDING state Sends/Receives STP BPDUs
- A PORT in the FORWARDING state **learns** MAC ADDRESSES

SUMMARY :

Spanning Tree Port States

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

STP TIMERS

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)

💡 SWITCHES do NOT forward the BPDUs out of their ROOT PORTS and NON-DESIGNATED PORTS - ONLY their DESIGNATED PORTS !!!

MAX AGE TIMER:

- If another BPDU is received BEFORE MAX AGE TIMER counts down to 0, the TIME will RESET to 20 Seconds and no changes will occur.
- If another BPDU is not received, the MAX AGE TIMER counts down to 0 and the SWITCH will re-evaluate it's STP choices, including ROOT BRIDGE, LOCAL ROOT, DESIGNATED, and NON-DESIGNATED PORTS.
- If a NON-DESIGNATED PORT is selected to become a DESIGNATED or ROOT PORT, it will transition from the BLOCKING state to the LISTENING state (15 Seconds), LEARNING state (15 Seconds), and then finally the FORWARDING state.
    - So… it can take 50 Seconds for a BLOCKING interface to transition to FORWARDING! (MAX AGE TIMER + (LISTENING + LEARNING 15 Second timers))
- These TIMERS and TRANSITIONAL STATES are to make sure that LOOPS are not accidentally created by an INTERFACE moving to FORWARDING STATE too soon

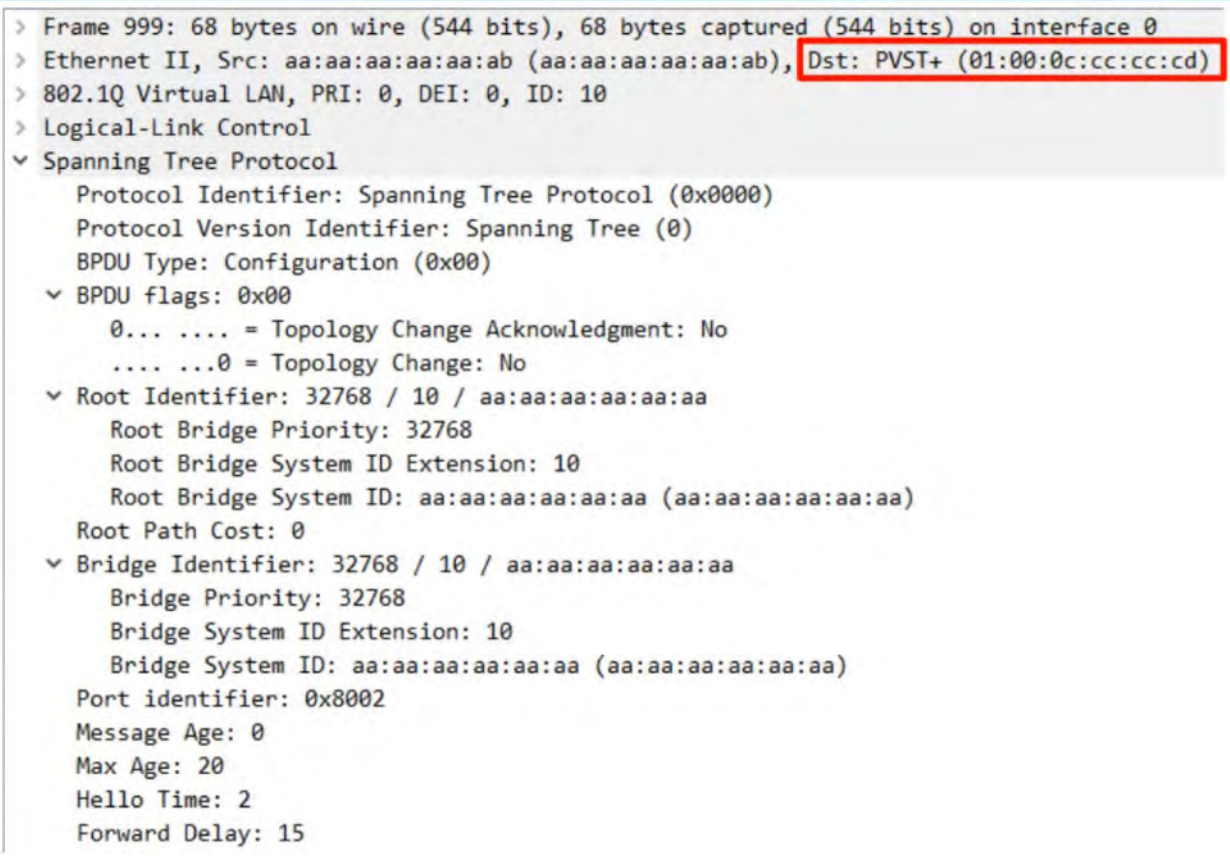
HOWEVER …

💡 A FORWARDING interface can move DIRECTLY to a BLOCKING state (there is no worry about creating a loop)

💡 A BLOCKING interface can NOT move DIRECTLY to a FORWARDING state. It MUST go through the LISTENING and LEARNING states first!

STP BPDU (BRIDGE PROTOCOL DATA UNIT)

Ethernet Header of a BPDU



💡 PVST+ uses the MAC ADDRESS :
01 : 00 : 0c : cc : cc : cd
PVST = ONLY ISL Trunk Encapsulation
PVST+ = Supports 802.1Q
💡 Regular STP (not Cisco's PVST+) uses the MAC ADDRESS :
01 : 80 : c2 : 00 : 00 : 00
💡 The STP TIMERS on the ROOT BRIDGE determine ALL STP TIMERS for the entire network!

STP OPTIONAL FEATURES (STP TOOLKIT)
PORTFAST:

- Can be Enabled on INTERFACES which are connected to END HOSTS

💡 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

```
SW1(config)#interface g0/2
SW1(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on GigabitEthernet0/2 but will only
 have effect when the interface is in a non-trunking mode.
SW1(config-if)#
```

You can also ENABLE PORTFAST with the following command:

💡 SW1(config)# spanning-tree portfast default

This ENABLES PORTFAST on ALL ACCESS PORTS (not TRUNK PORTS)

BPDU GUARD:

- If an INTERFACE with BPDU GUARD ENABLED receives a BPDU from another SWITCH, the INTERFACE will be SHUT DOWN to prevent loops from forming.

```
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

ROOT GUARD / LOOP GUARD:

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.

You probably do NOT have to know these STP optional features (or others such as UplinkFast, Backbone Fast, etcetera) for the CCNA.

BUT…

💡 Make sure you know PORTFAST and BPDU GUARD.

STP CONFIGURATION

Command to CONFIGURE Spanning-Tree mode on a SWITCH

```
SW1(config)#spanning-tree mode ?
  mst         Multiple spanning tree mode
  pvst        Per-Vlan spanning tree mode
  rapid-pvst  Per-Vlan rapid spanning tree mode

SW1(config)#spanning-tree mode pvst
```

Modern Cisco SWITCHES run **rapid-pvst**, by default

CONFIGURE THE PRIMARY ROOT BRIDGE

Command to CONFIGURE Spanning-Tree PRIMARY ROOT BRIDGE on a SWITCH

```
SW3(config)#spanning-tree vlan 1 root primary
SW3(config)#do show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     cccc.cccc.cccc
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    24577  (priority 24576 sys-id-ext 1)
              Address     cccc.cccc.cccc
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  15  sec
```

Confirm with "(do) show spanning-tree"
Can see in the above example, SW3 has become the "root"

- The "spanning-tree vlan root primary" command sets the STP PRIORITY to 24576. If another SWITCH already has a priority number lower than 24576, it sets this SWITCH's priority to 4096 LESS THAN the other SWITCH's Priority (remember STP PART 1 lecture)

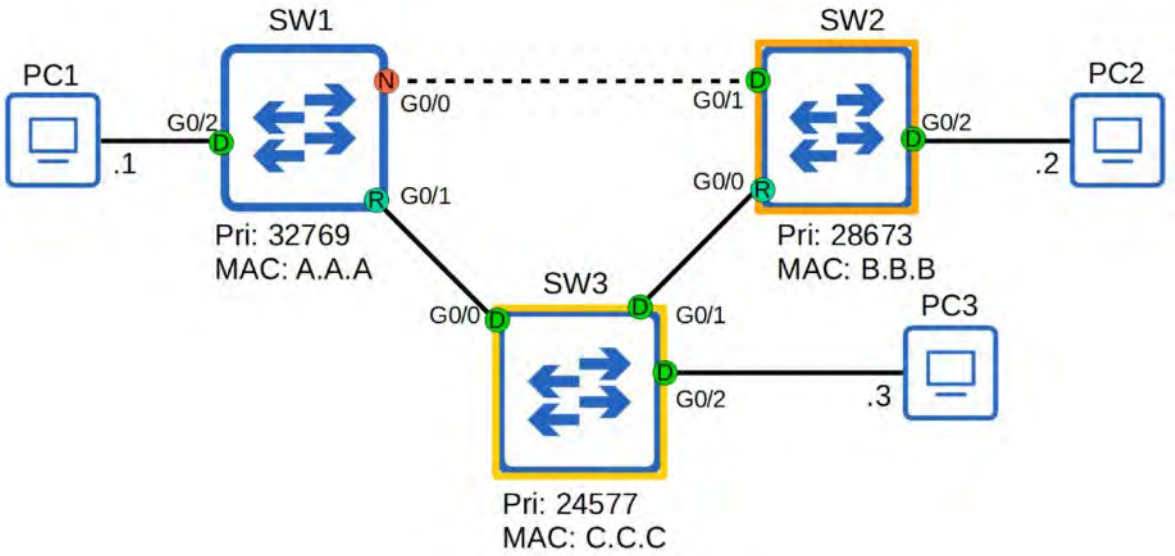
SECONDARY ROOT BRIGE (backup ROOT BRIDGE)
Command to CONFIGURE Spanning-Tree SECONDARY ROOT BRIDGE on a SWITCH

```
SW2(config)#spanning-tree vlan 1 root secondary
SW2(config)#do show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    24577
              Address     cccc.cccc.cccc
              Cost        4
              Port        1 (GigabitEthernet0/0)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28673  (priority 28672 sys-id-ext 1)
              Address     bbbb.bbbb.bbbb
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec
```

- The "spanning-tree vlan root secondary" command sets the STP PRIORITY to 28672 (exactly 4096 higher than 24576).

VLAN 1 TOPOLOGY running PVST+

STP Load-Balancing

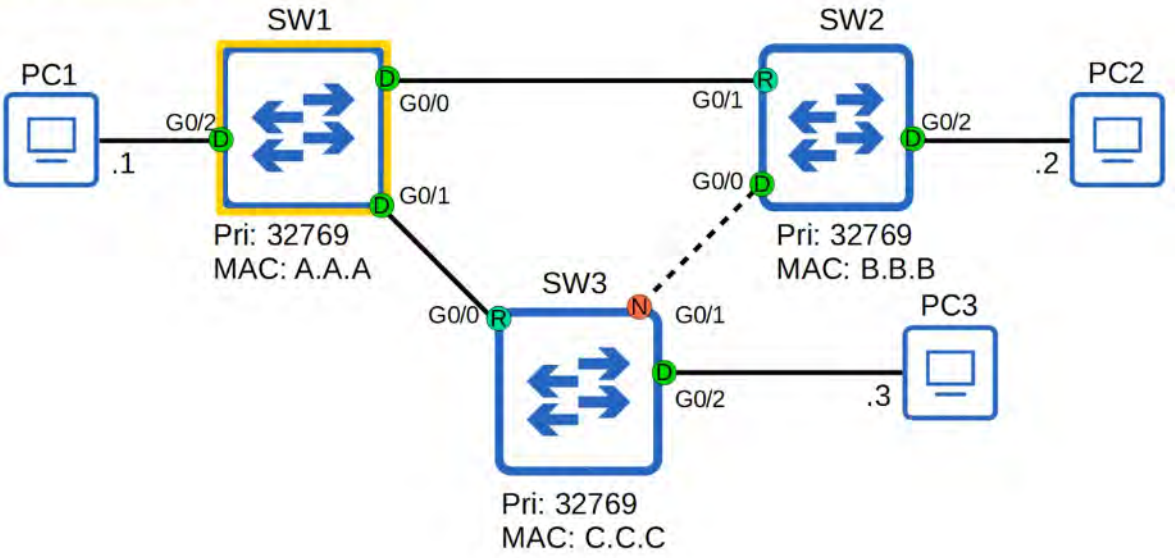


SW1 WAS the PRIMARY ROOT BRIDGE but :

- We have configured SW3 to be the PRIMARY
- We have configured SW2 to be the SECONDARY

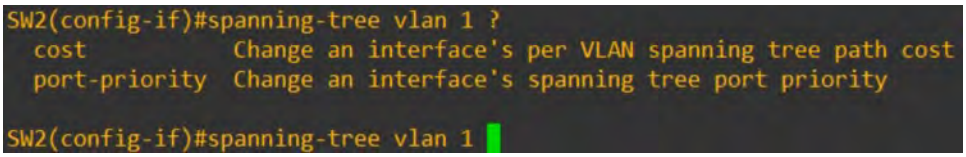
The TOPOLOGY for VLAN 2, however, won't be the same. It will be the OLD Topology.

VLAN 2 Topology



WHY? Because we made changes ONLY to the TOPOLOGY found in VLAN 1 (see the commands we used)

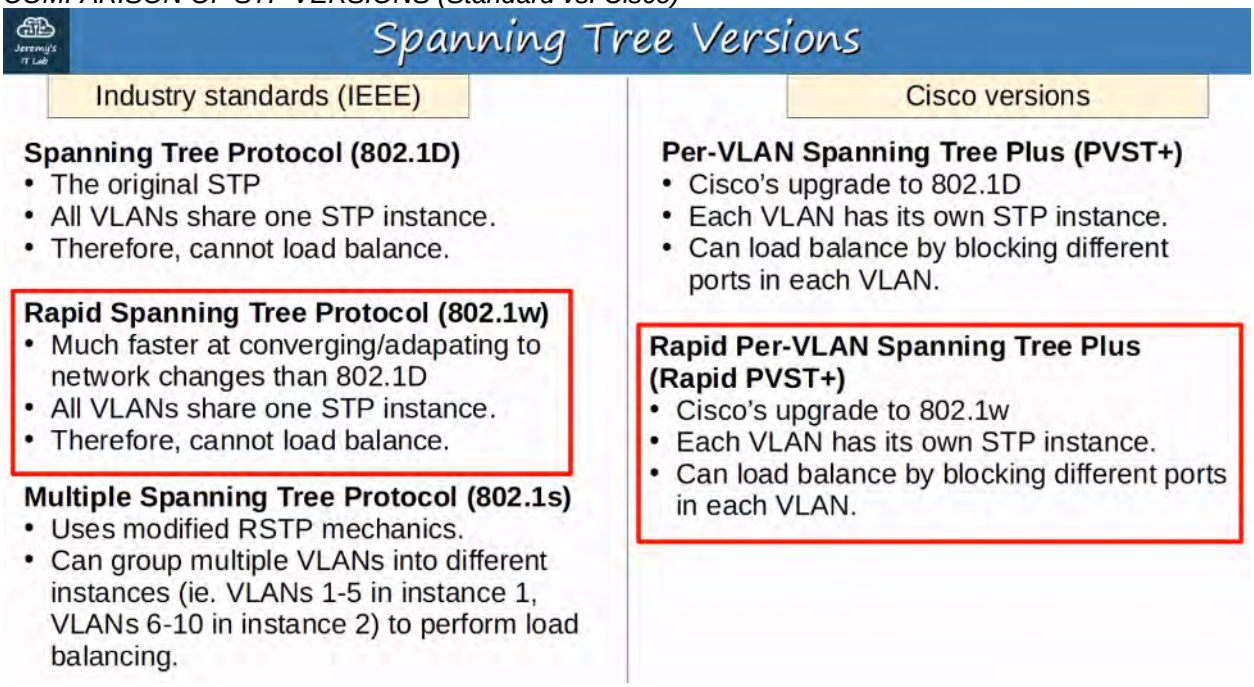
CONFIGURE STP PORT SETTINGS



“cost” = “ROOT COST”
“port-priority” = “PORT PRIORITY”

22. RAPID SPANNING TREE PROTOCOL

COMPARISON OF STP VERSIONS (Standard vs. Cisco)



We are only concerned with 802.1w for MOST use cases.
MSTP (802.1s) is more useful for VERY LARGE networks.
WHAT IS RAPID PER-VLAN SPANNING TREE PLUS?
RSTP is not a time-based spanning tree algorithm like 802.1D. Therefore, RSTP offers an improvment over teh 30 seconds or more 802.1D takes to move a link to forwarding. The heart of the protocol is a new bridge-bridge handshake mechanism, which allows ports to move directly to forwarding

---

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

DIFFERENCES BETWEEN STP AND RSTP:

PORT COSTS

Speed	STP Cost	RSTP Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2000
100 Gbps	X	200
1 Tbps	X	20

(STUDY AND MEMORIZE PORT COSTS OF STP AND RSTP)
RSTP PORT STATES

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

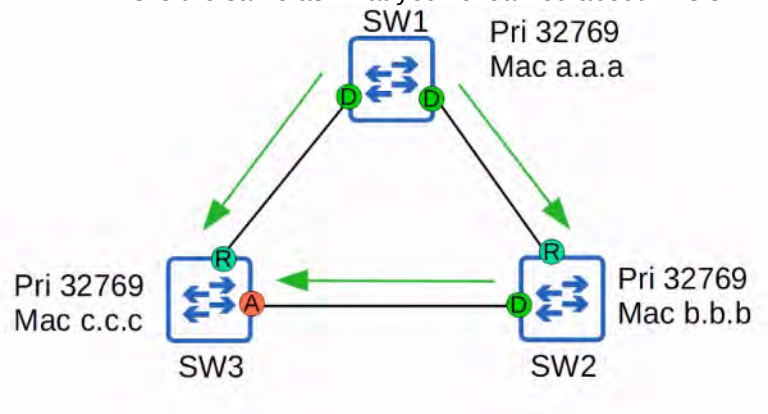
- If a PORT has been ADMINISTRATIVELY DISABLED (”shutdown” command) = DISCARDING STATE
- If a PORT is ENABLED but BLOCKING traffic to prevent LAYER 2 LOOPS = DISCARDING STATE

RSTP ROLES

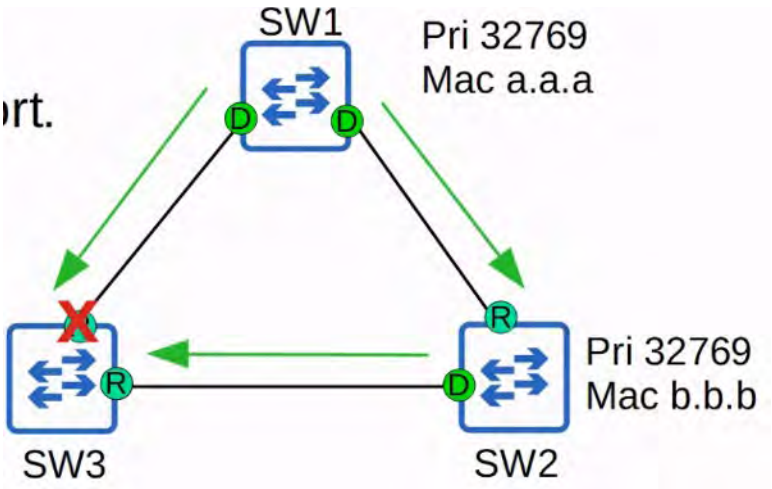
- The ROOT PORT role remains unchanged in RSTP
  - The PORT that is closest to the ROOT BRIDGE becomes the ROOT PORT for the SWITCH
  - The ROOT BRIDGE is the only SWITCH that doesn’t have a ROOT PORT
- The DESIGNATED PORT role remains unchanged in RSTP
  - The PORT on a segment (Collision Domain) that 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 : ALTERNATE PORT ROLE

- The RSTP ALTERNATE PORT ROLE is a DISCARDING PORT that receives a superior BPDU from another SWITCH
- This is the same as what you’ve learned about BLOCKING PORTS in classic STP

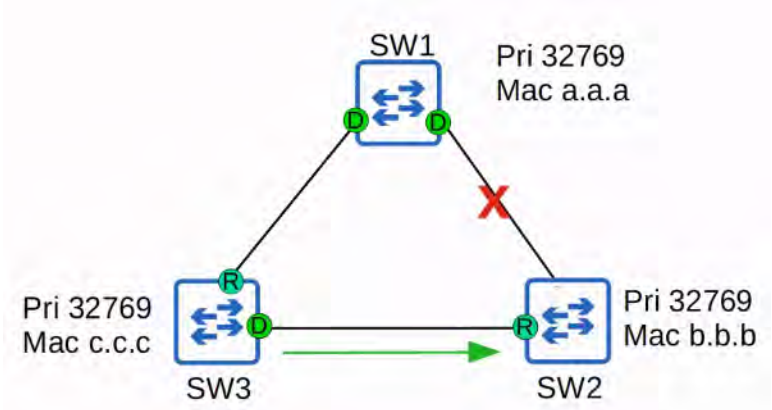


- An ALTERNATE PORT (labelled “A” above) functions as a backup to the ROOT PORT
- If the ROOT PORT fails, the SWITCH can immediately move it’s best alternate port to FORWARDING



💡 This immediate move to FORWARDING STATE functions like a classic STP optional feature called **UplinkFast.** Because it is built into RSTP, you do not need to activate UplinkFast when using RSTP/Rapid PVST+

One more STP optional feature that was built into RSTP is **BackboneFast**



- **BackboneFast** allows SW3 to expire the MAX AGE TIMERS on it's INTERFACE and rapidly FORWARD the superior BPDUs to SW2
- This FUNCTIONALITY is built into RSTP, so it does not need to be configured.

UPLINKFAST and BACKBONE FAST (SUMMARY)

💡 **UplinkFast** and **BackboneFast** are two optional features in Classic STP. They must be configured to operate on the SWITCH (not necessary to know for the CCNA)

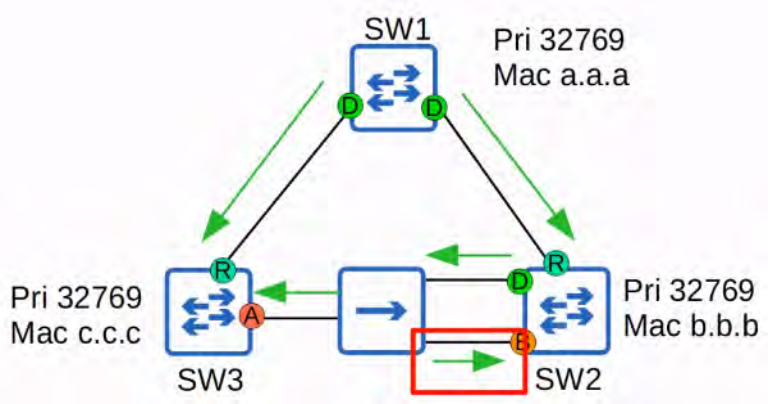
- Both features are built into RSTP, so you do NOT have to configure them. They operate, by DEFAULT
- You do NOT need to have a detailed understanding of them for the CCNA. Know their names and their BASIC purpose (to help BLOCKING / DISCARDING PORTS rapidly move to FORWARDING)

RSTP : BACKUP PORT ROLE

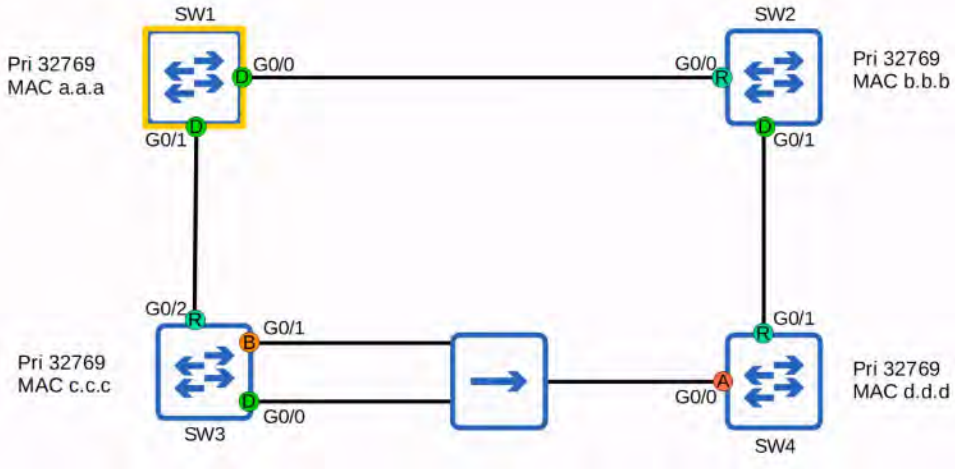
- The RSTP BACKUP PORT role is a DISCARDING PORT that receives a superior BPDU from another INTERFACE on the same SWITCH
- This only happens when TWO INTERFACES are connected to the SAME COLLISION DOMAIN (via a HUB)
- Hubs are NOT used in modern networks, so you will probably NOT encounter an RSTP BACKUP PORT
- Hubs are NOT used in modern networks, so you will probably NOT encounter an RSTP BACKUP PORT.

- Functions as a BACKUP for a DESIGNATED PORT

💡 The INTERFACE with the LOWERS PORT ID will be selected as the DESIGNATED PORT, and the other will be the BACKUP port.



WHICH Switch will be ROOT BRIDGE? What about the OTHER ports ?

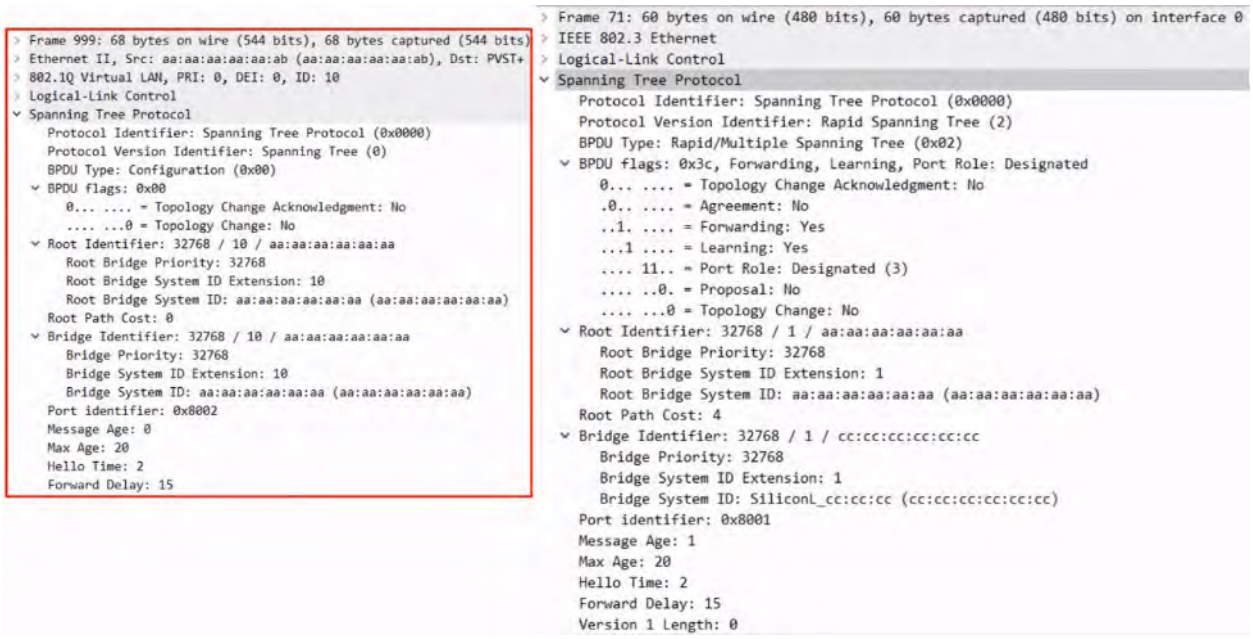


```
SW3(config)#spanning-tree mode ?
  mst         Multiple spanning tree mode
  pvst        Per-Vlan spanning tree mode
  rapid-pvst  Per-Vlan rapid spanning tree mode

SW3(config)#spanning-tree mode rapid-pvst
SW3(config)#do show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    32769
              Address     aaaa.aaaa.aaaa
              Cost        4
              Port        3 (GigabitEthernet0/2)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     cccc.cccc.cccc
              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               Desg FWD 4         128.1    Shr
Gi0/1               Back BLK 4         128.2    Shr
Gi0/2               Root FWD 4         128.3    P2p
```

```
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, etc.)

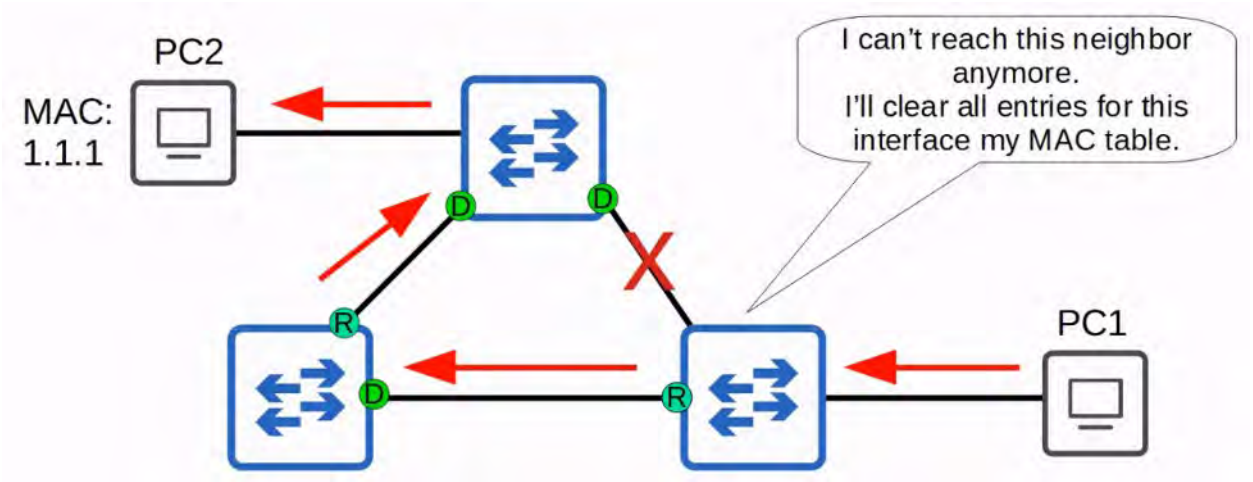
RAPID STP BPDU
CLASSIC RSTP (LEFT) vs RAPID STP BPDU (RIGHT)



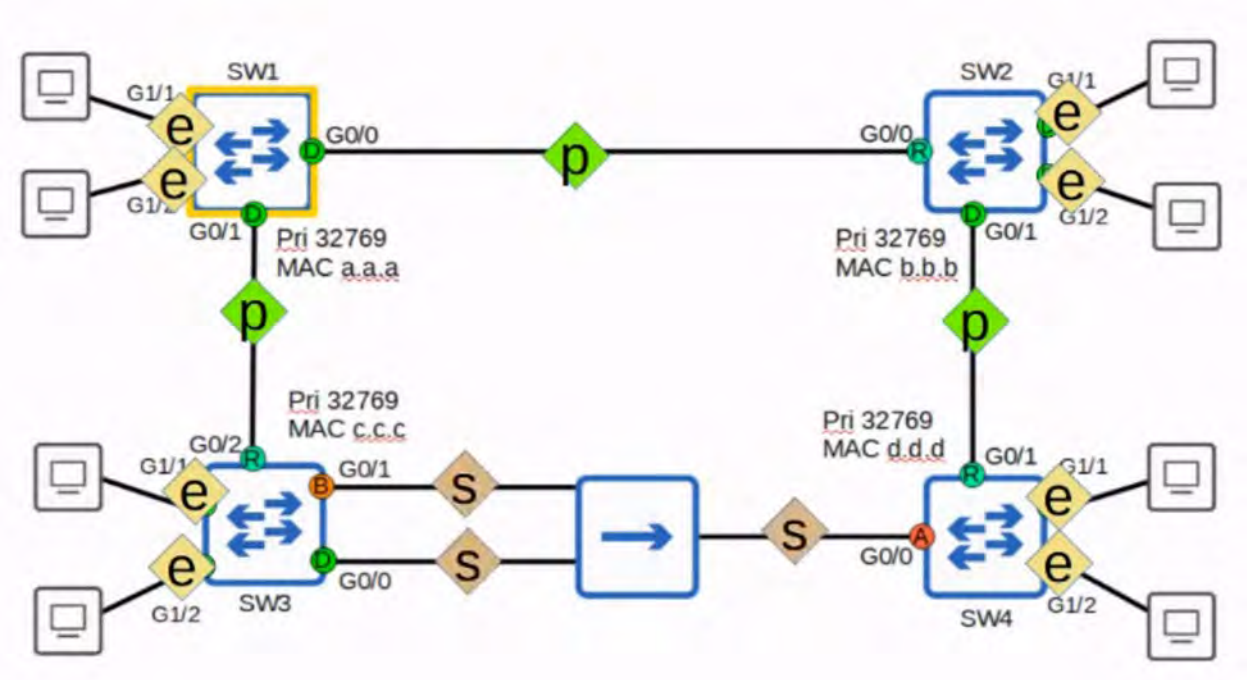
💡 NOTE:
Classic STP BPDU has a “Protocol Version Identifier: Spanning Tree (0)
BPDU Type: Configuration (0x00)
BPDU flags: 0x00
RAPID STP BPDU has a “Protocol Version Identifier: Spanning Tree (2)
BPDU Type: Configuration (0x02)
BPDU flags: 0x3c
In CLASSIC STP, only the ROOT BRIDGE originated BPDUs, and other SWITCHES just FORWARDED the BPDUs they received.
In RAPID STP, ALL SWITCHES originate and send their own BPDUs from their DESIGNATED PORTS

RAPID SPANNING TREE PROTOCOL

- ALL SWITCHES running RAPID STP send their own BPDUs every “hello” time (2 Seconds)
- SWITCHES “age” the BPDU information much more quickly
    - In CLASSIC STP, a SWITCH waits 10 “hello” intervals (20 seconds)
    - In RAPID STP, a SWITCH considers a neighbour lost if it misses 3 BPDUs (6 seconds). It will then “flush” ALL MAC ADDRESSES learned on that interface



RSTP LINK TYPES



<E> = EDGE

<P> = POINT-TO-POINT

<S> = SHARED

RSTP distinguishes between THREE different “link types” : **EDGE**, **POINT-TO-POINT**, and **SHARED**

EDGE PORTS

- Connected to END HOSTS
- Because there is NO RISK of creating a LOOP, they can move straight to the FORWARDING STATE without the negotiation process!
- They function like a CLASSIC STP PORT with PORTFAST ENABLED

💡 SW1(config-if)# spanning-tree portfast

POINT-TO-POINT PORTS

- Connect directly to another SWITCH
- They function in FULL-DUPLEX
- You don’t need to configure the INTERFACE as POINT-TO-POINT (it should be detected)

💡 SW1(config-if)# spanning-tree link-type point-to-point

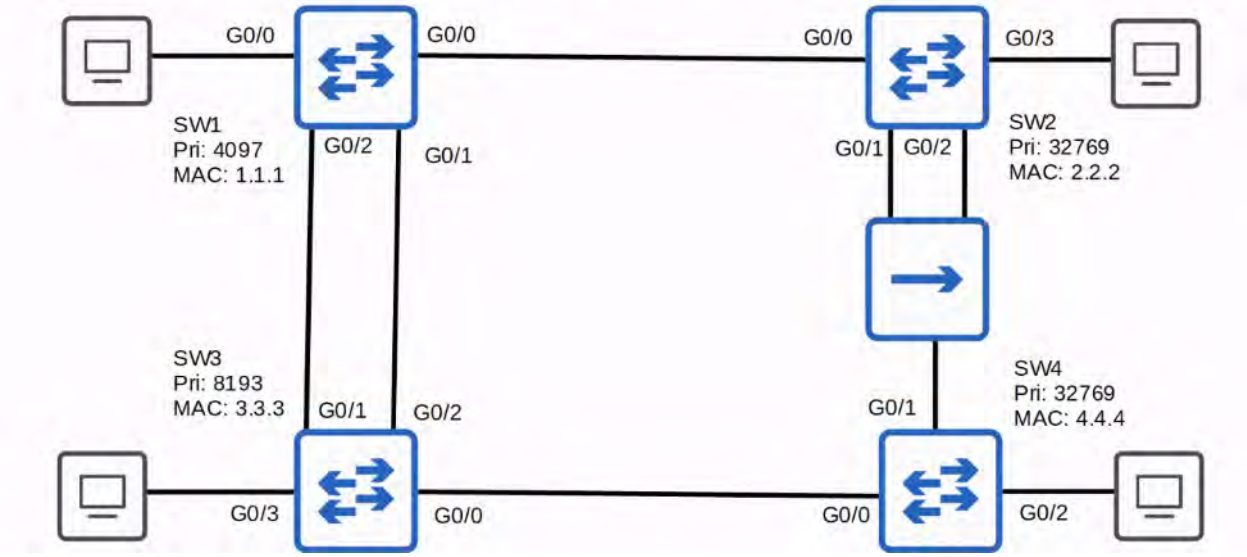
SHARED PORTS

- Connect to another SWITCH (or SWITCHES) via a HUB
- They function in HALF-DUPLEX
- You don’t need to configure the INTERFACE as SHARED (it should be detected)

💡 SW1(config-if)# spanning-tree link-type shared

QUIZ:

Identify the root bridge in this network. What is the RSTP port role of each switch port? What is the appropriate RSTP link type of each connection between devices?



SW1 :

- **ROOT BRIDGE**
- G0/0 - 0/3= DESIGNATED

SW2 :

- G0/0 = ROOT PORT
- G0/1 = DESIGNATED PORT
- G0/2 = BACKUP PORT
- G0/3 = DESIGNATED PORT

SW3 :

- G0/0 = DESIGNATED PORT
- G0/1 = ALTERNATE PORT
- G0/2 = ROOT PORT
- G0/3 = DESIGNATED PORT

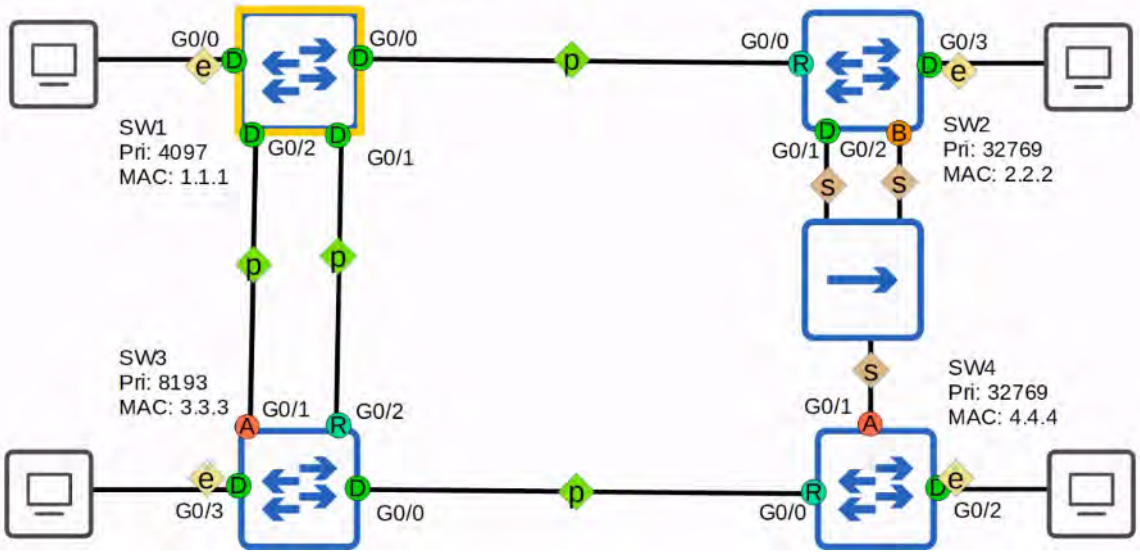
SW4:

- G0/0 = ROOT
- G0/1 = ALTERNATE PORT
- G0/2 = DESIGNATED PORT

Connection between SW1 G0/0 and SW2 G0/0 = POINT-TO-POINT
Connection between SW3 G0/0 and SW4 G0/0 = POINT-TO-POINT
Connection between SW1 G0/1 and G0/2 to SW3 G0/1 and G0/2 = POINT-TO-POINT
Connections to all the END HOSTS = EDGE
Connection from SW4 to HUB = SHARED
Connections from SW2 to HUB = SHARED
ANSWER

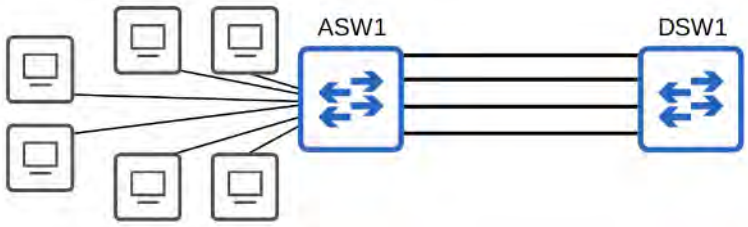
RSTP Quiz 4

Identify the root bridge in this network. What is the RSTP port role of each switch port? What is the appropriate RSTP link type of each connection between devices?

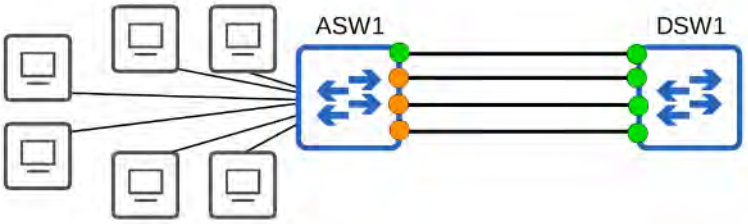


23. ETHERCHANNEL

WHAT IS ETHERCHANNEL?
ETHERCHANNEL allows you to GROUP multiple physical INTERFACES into a group which operates as a SINGLE LOGICAL INTERFACE - so they BEHAVE as if they are a single INTERFACE
A LAYER 2 ETHERCHANNEL is a group of SWITCH PORTS which operate as a SINGLE INTERFACE
A LAYER 3 ETHERCHANNEL is a group of ROUTED PORTS which operate as a SINGLE INTERFACE which you assign an IP ADDRESS to.



When the bandwidth of the INTERFACES connected to END HOSTS is greater than the bandwidth of the connection to the DISTRIBUTION SWITCH(es), this is called **OVERSUBSCRIPTION.**
Some OVERSUBSCRIPTION is acceptable, but too much will cause congestion.

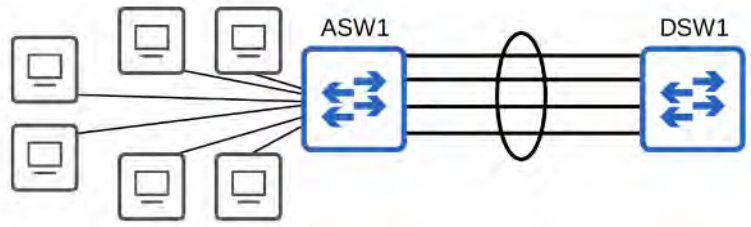


- If you connect TWO SWITCHES together with multiple links, ALL except ONE will be DISABLED by SPANNING TREE PROTOCOL (Green Lights vs. Orange Lights above on ASW1)

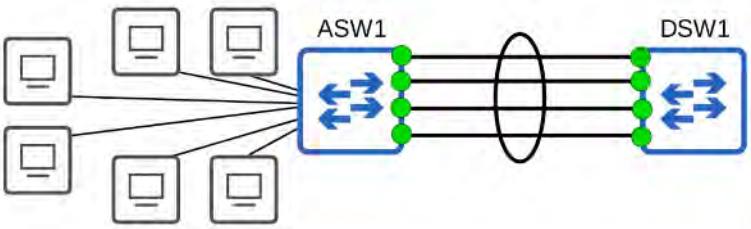
WHY?

- If ALL of ASW1s INTERFACES were FORWARDING, LAYER 2 LOOPS would form between ASW1 and DSW1, leading to a BROADCAST STORM (Bad!)
- Other links will be unused unless the active link fails. In that case, one of the inactive link will start forwarding.

An ETHERCHANNEL (in network topology diagrams) is represented like THIS (circle around multi-connections)



- ETHERCHANNEL groups multiple channels together to act as a SINGLE INTERFACE
- STP will treat this GROUP as a SINGLE INTERFACE

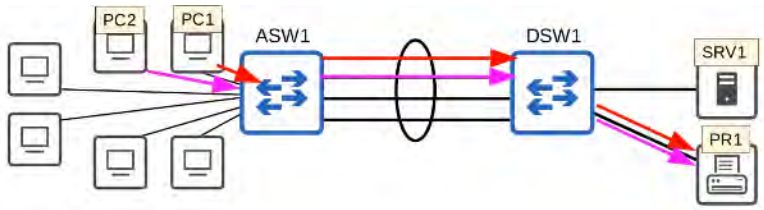


(All INTERFACES Green!)
TRAFFIC using ETHERCHANNEL will be load-balanced among the physical INTERFACES in the group.

An algorithm is used to determine WHICH TRAFFIC will use WHICH physical INTERFACE (more details later)
Some other names for an ETHERCHANNEL are:

- PORT CHANNEL
- LAG (Link Aggregation Group)

HOW DOES AN ETHERCHANNEL LOAD-BALANCE?



- ETHERCHANNEL load-balances based on **"flows"**
- A "flow" is a communication between TWO NODES in the NETWORK
- FRAMES in the same "flow" will be FORWARDED using the SAME physical INTERFACE
- If FRAMES in the same "flow" were FORWARDED using different physical INTERFACES, some FRAMES may arrive at the DESTINATION out of order/sequence, which can cause problems.
- You can CHANGE the INPUTS used in the INTERFACE SELECTION calculation (for "flows")
    - INPUTS that can be used:
        - SOURCE MAC ADDRESS
        - DESTINATION MAC ADDRESS
        - SOURCE and DESTINATION MAC ADDRESS
        - SOURCE IP ADDRESS
        - DESTINATION IP ADDRESS
        - SOURCE and DESTINATION IP ADDRESS

How to see the configuration for LOAD-BALANCING method

```
ASW1#show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination IP address
  IPv6: Source XOR Destination IP address
```

How to CHANGE the LOAD-BALANCING method

```
ASW1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ASW1(config)#port-channel load-balance src-dst-mac
ASW1(config)#do show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
        src-dst-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
  IPv4: Source XOR Destination MAC address
  IPv6: Source XOR Destination MAC address

ASW1(config)#
```

```
ASW1(config)#port-channel load-balance ?
  dst-ip        Dst IP Addr
  dst-mac       Dst Mac Addr
  src-dst-ip    Src XOR Dst IP Addr
  src-dst-mac   Src XOR Dst Mac Addr
  src-ip        Src IP Addr
  src-mac       Src Mac Addr

ASW1(config)#port-channel load-balance
```

HOW TO CONFIGURE LAYER 2 / LAYER 3 ETHERCHANNELS
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 in STANDBY mode, waiting for an active INTERFACE to fail)

PAgP CONFIGURATION

```
ASW1(config)#interface range g0/0 - 3
ASW1(config-if-range)#channel-group 1 mode ?
  active     Enable LACP unconditionally
  auto       Enable PAgP only if a PAgP device is detected
  desirable  Enable PAgP unconditionally
  on         Enable Etherchannel only
  passive    Enable LACP only if a LACP device is detected

ASW1(config-if-range)#channel-group 1 mode desirable
Creating a port-channel interface Port-channel 1
```

💡 NOTE that "auto" and "desirable" are the ONLY available modes for PAgP

auto + auto = no EtherChannel
desirable + auto = EtherChannel
desirable + desirable = EtherChannel

PAgP negotiations to form an ETHERCHANNEL

💡 AWS1(config-if-range)# channel-group 1 mode desirable. Creating a port-channel interface Port-channel1
Shows up in the interface as "Port-channel1"

```
ASW1(config)#do show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES unset  up                    up
GigabitEthernet0/1     unassigned      YES unset  up                    up
GigabitEthernet0/2     unassigned      YES unset  up                    up
GigabitEthernet0/3     unassigned      YES unset  up                    up
GigabitEthernet1/0     unassigned      YES unset  up                    up
GigabitEthernet1/1     unassigned      YES unset  up                    up
GigabitEthernet1/2     unassigned      YES unset  up                    up
GigabitEthernet1/3     unassigned      YES unset  up                    up
GigabitEthernet2/0     unassigned      YES unset  up                    up
GigabitEthernet2/1     unassigned      YES unset  up                    up
GigabitEthernet2/2     unassigned      YES unset  up                    up
GigabitEthernet2/3     unassigned      YES unset  up                    up
GigabitEthernet3/0     unassigned      YES unset  up                    up
GigabitEthernet3/1     unassigned      YES unset  up                    up
GigabitEthernet3/2     unassigned      YES unset  up                    up
GigabitEthernet3/3     unassigned      YES unset  up                    up
Port-channel1          unassigned      YES unset  up                    up
ASW1(config)#
```

The “channel-group” number has to MATCH for member INTERFACES on the same SWITCH.
It DOESN’T have to MATCH the “channel-group” number on the OTHER SWITCH!

💡 (channel-group 1 on AWS1 can form an ETHERCHANNEL with channel-group 2 on DSW1)

LACP CONFIGURATION

```
ASW1(config-if-range)#channel-group 1 mode ?
  active     Enable LACP unconditionally
  auto       Enable PAgP only if a PAgP device is detected
  desirable  Enable PAgP unconditionally
  on         Enable Etherchannel only
  passive    Enable LACP only if a LACP device is detected

ASW1(config-if-range)#channel-group 1 mode active
Creating a port-channel interface Port-channel 1
```

💡 NOTE that “active” and “passive” are the ONLY available modes for LACP

passive + passive = no EtherChannel
active + passive = EtherChannel
active + active = EtherChannel

LACP negotiations for form an ETHERCHANNEL
The “channel-group” number has to MATCH for member INTERFACES on the same SWITCH.
It DOESN’T have to MATCH the “channel-group” number on the OTHER SWITCH!

💡 (channel-group 1 on AWS1 can form an ETHERCHANNEL with channel-group 2 on DSW1)

STATIC ETHERCHANNEL CONFIGURATION

```
ASW1(config-if-range)#channel-group 1 mode ?
  active      Enable LACP unconditionally
  auto        Enable PAgP only if a PAgP device is detected
  desirable   Enable PAgP unconditionally
  on          Enable Etherchannel only
  passive     Enable LACP only if a LACP device is detected

ASW1(config-if-range)#channel-group 1 mode on
Creating a port-channel interface Port-channel 1
```

💡 NOTE that “on” is the ONLY available mode for STATIC ETHERCHANNEL
ON mode only works with ON Mode
ON + desirable = DOES NOT WORK)
ON + active = DOES NOT WORK

HOW TO MANUALLY CONFIGURE THE NEGOTIATION PROTOCOL

```
ASW1(config-if-range)#channel-protocol ?
  lacp  Prepare interface for LACP protocol
  pagp  Prepare interface for PAgP protocol

ASW1(config-if-range)#channel-protocol lacp
ASW1(config-if-range)#channel-group 1 mode desirable
Command rejected (Channel protocol mismatch for interface Gi0/0 in group 1): the interface can not be added to the channel group

% Range command terminated because it failed on GigabitEthernet0/0
ASW1(config-if-range)#channel-group 1 mode on
Command rejected (Channel protocol mismatch for interface Gi0/0 in group 1): the interface can not be added to the channel group

% Range command terminated because it failed on GigabitEthernet0/0
ASW1(config-if-range)#channel-group 1 mode active
Creating a port-channel interface Port-channel 1

ASW1(config-if-range)#
```

TWO OPTIONS:

- LACP Protocol
- PAgP Protocol

(Above shows a protocol mismatch error because LACP does not support “desirable” - only PAgP does)
(“channel-group 1 mode active” works because LACP supports “active”)

AFTER CONFIGURING THE ETHERCHANNEL MODE
CONFIGURING THE PORT INTERFACE

```
ASW1(config)#interface port-channel 1
ASW1(config-if)#switchport trunk encapsulation dot1q
ASW1(config-if)#switchport mode trunk
ASW1(config-if)#do show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Po1         on               802.1q         trunking      1

Port        Vlans allowed on trunk
Po1         1-4094

Port        Vlans allowed and active in management domain
Po1         1

Port        Vlans in spanning tree forwarding state and not pruned
Po1         none
```

“show running-config” shows “interface Port-channel1” in the configuration

```
interface Port-channel1
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface GigabitEthernet0/0
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
 channel-protocol lacp
 channel-group 1 mode active
!
interface GigabitEthernet0/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
 channel-protocol lacp
 channel-group 1 mode active
!
interface GigabitEthernet0/2
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
 channel-protocol lacp
 channel-group 1 mode active
!
interface GigabitEthernet0/3
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
 channel-protocol lacp
 channel-group 1 mode active
!
```

💡 NOTE the PHYSICAL INTERFACES (g0/0-g0/3) were auto-configured by the Port-channel1 configuration!

IMPORTANT NOTES ABOUT ETHERCHANNEL CONFIGURATION

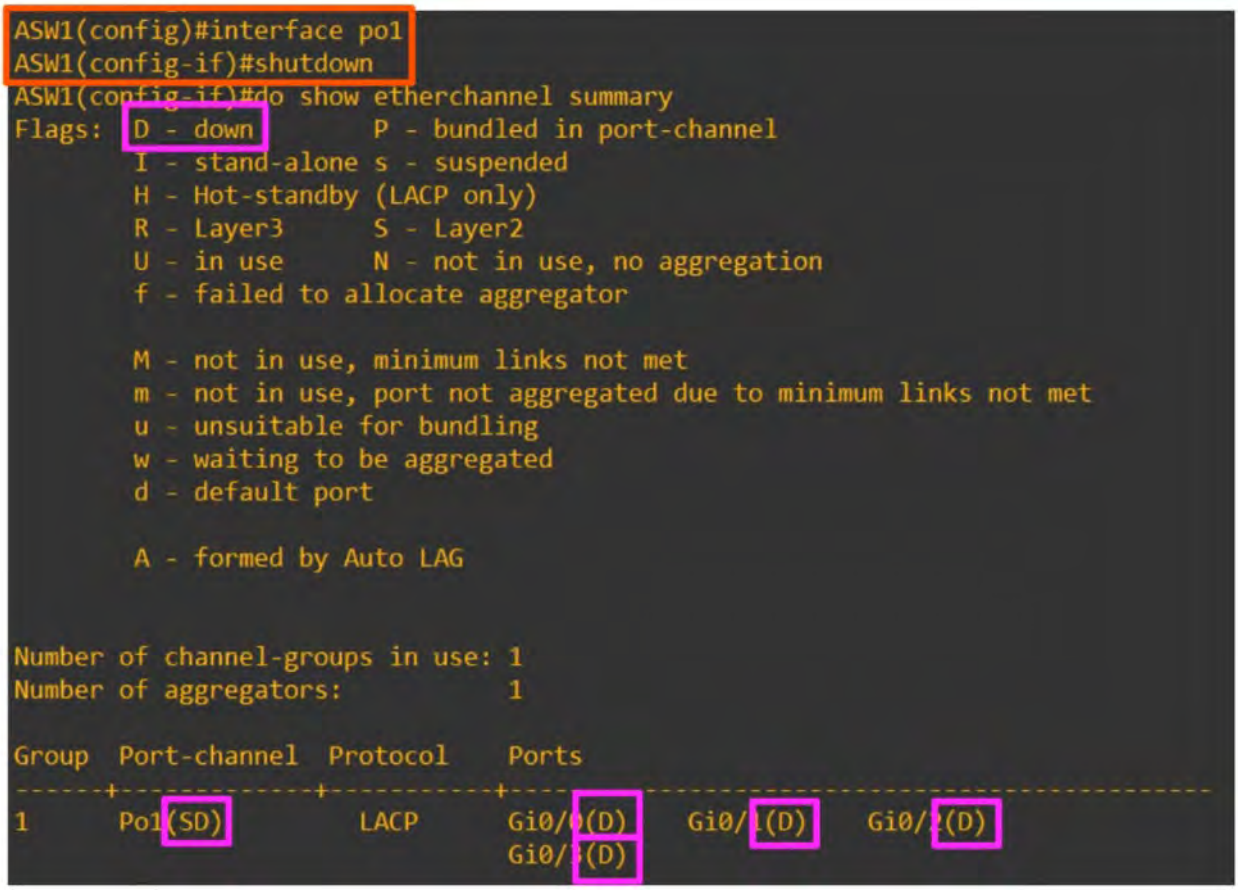
- Member INTERFACES must have matching CONFIGURATIONS
    - Same DUPLEX (Full / Half)
    - Same SPEED
    - Same SWITCHPORT mode (Access / Trunk)
    - Same allowed VLANs / Native VLAN (for TRUNK interfaces)
- If an INTERFACE's configurations do NOT MATCH the others, it will be EXCLUDED from the ETHERCHANNEL

VERIFYING STATUS OF ETHERCHANNEL

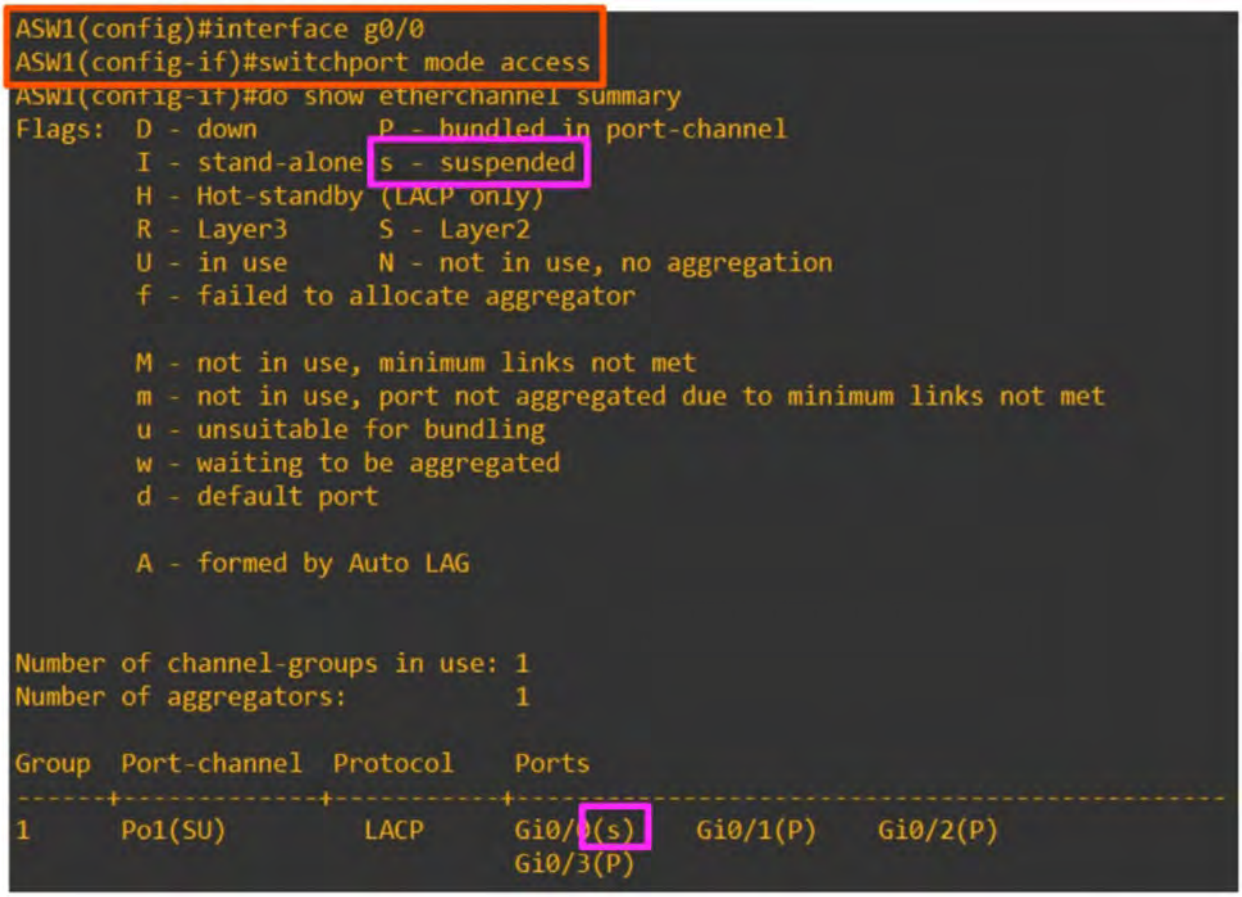
💡 "show etherchannel summary"

```
ASW1#show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         LACP      Gi0/0(P)    Gi0/1(P)    Gi0/2(P)
                                 Gi0/3(P)
```

NOTE information at bottom. (”SU” means S - Layer2 + U - in use)
Protocol = What protocol the Etherchannel is using (in this case, LACP)
“Ports” = the list of interfaces in the EtherChannel (P = bundled in port-channel)
OTHER FLAGS


ASW1(config)#interface po1
ASW1(config-if)#shutdown
ASW1(config-if)#do show etherchannel summary
Flags: D - down P - bundled in port-channel
I - stand-alone s - suspended
H - Hot-standby (LACP only)
R - Layer3 S - Layer2
U - in use N - not in use, no aggregation
f - failed to allocate aggregator
M - not in use, minimum links not met
m - not in use, port not aggregated due to minimum links not met
u - unsuitable for bundling
w - waiting to be aggregated
d - default port
A - formed by Auto LAG
Number of channel-groups in use: 1
Number of aggregators: 1
Group Port-channel Protocol Ports
1 Po1(SD) LACP Gi0/0(D) Gi0/1(D) Gi0/2(D)
Gi0/3(D)

“D” = Down



Changing one of the Member interfaces using “switchport mode access” has made it different than the other members so it is now appearing as “s” = suspended

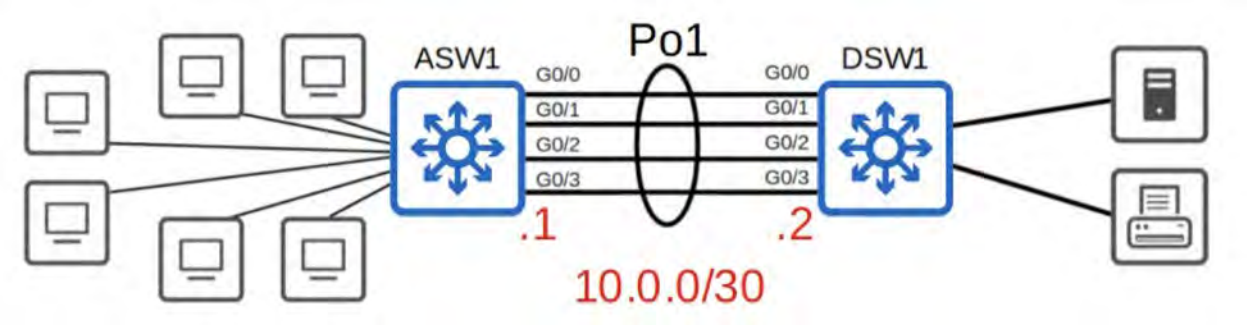
Another useful command

💡 “show etherchannel port-channel”

```
ASW1#show etherchannel port-channel
                Channel-group listing:
                ----------------------

Group: 1
----------
                Port-channels in the group:
                ---------------------------

Port-channel: Po1    (Primary Aggregator)

------------

Age of the Port-channel   = 0d:00h:36m:48s
Logical slot/port   = 16/0          Number of ports = 4
HotStandBy port = null
Port state          = Port-channel Ag-Inuse
Protocol            =   LACP
Port security       = Disabled

Ports in the Port-channel:

Index   Load   Port     EC state        No of bits
------+------+------+------------------+-----------
  0     00     Gi0/0    Active             0
  0     00     Gi0/1    Active             0
  0     00     Gi0/2    Active             0
  0     00     Gi0/3    Active             0

Time since last port bundled:    0d:00h:00m:02s    Gi0/0
Time since last port Un-bundled: 0d:00h:08m:42s    Gi0/0
```

💡 “show spanning-tree” will show the single EtherChannel port interface

```
ASW1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    32769
              Address     0c04.cf10.ea00
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     0c04.cf10.ea00
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Po1                 Desg FWD 3         128.65   Shr
```



HOW TO CONFIGURE A LAYER 3 ETHERCHANNEL (from a clean configuration)

```
ASW1(config)#int range g0/0 - 3
ASW1(config-if-range)#no switchport
ASW1(config-if-range)#channel-group 1 mode active
Creating a port-channel interface Port-channel 1
```

💡 “show running-config”

```
interface Port-channel1
 no switchport
 no ip address
!
interface GigabitEthernet0/0
 no switchport
 no ip address
 negotiation auto
 channel-group 1 mode active
!
interface GigabitEthernet0/1
 no switchport
 no ip address
 negotiation auto
 channel-group 1 mode active
!
interface GigabitEthernet0/2
 no switchport
 no ip address
 negotiation auto
 channel-group 1 mode active
!
interface GigabitEthernet0/3
 no switchport
 no ip address
 negotiation auto
 channel-group 1 mode active
!
```

NOTE : No SWITCHPORT and NO IP INTERFACE.

Where do we configure the IP Address? Directly on the PORT INTERFACE !

```
ASW1(config-if-range)#int po1
ASW1(config-if)#ip address 10.0.0.1 255.255.255.252
ASW1(config-if)#
```

```
ASW1(config-if)#do sh etherch sum
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

        A - formed by Auto LAG


Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(RU)         LACP      Gi0/0(P)    Gi0/1(P)    Gi0/2(P)
                                 Gi0/3(P)
```

(”RU” - “R” = Layer 3, “U” = in use)

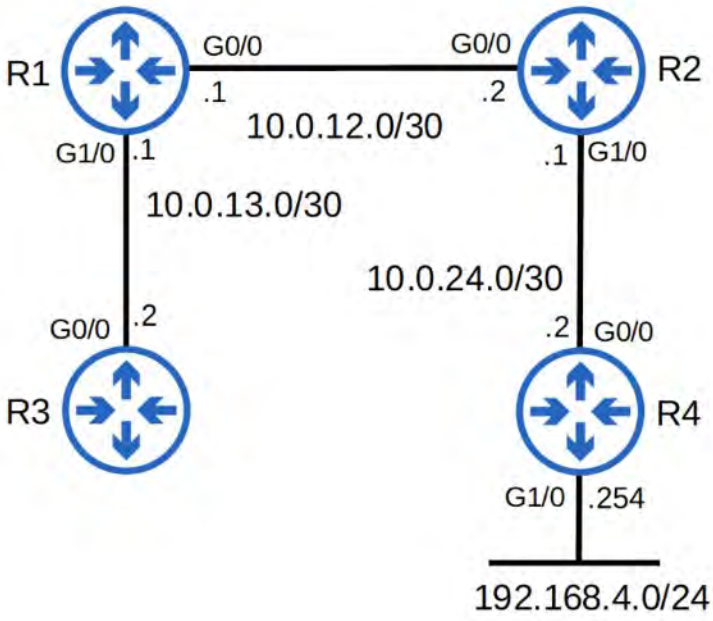
```
ASW1#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES manual up                    up
GigabitEthernet0/1     unassigned      YES manual up                    up
GigabitEthernet0/2     unassigned      YES manual up                    up
GigabitEthernet0/3     unassigned      YES manual up                    up
GigabitEthernet1/0     unassigned      YES unset  up                    up
GigabitEthernet1/1     unassigned      YES unset  up                    up
GigabitEthernet1/2     unassigned      YES unset  up                    up
GigabitEthernet1/3     unassigned      YES unset  up                    up
GigabitEthernet2/0     unassigned      YES unset  up                    up
GigabitEthernet2/1     unassigned      YES unset  up                    up
GigabitEthernet2/2     unassigned      YES unset  up                    up
GigabitEthernet2/3     unassigned      YES unset  up                    up
GigabitEthernet3/0     unassigned      YES unset  up                    up
GigabitEthernet3/1     unassigned      YES unset  up                    up
GigabitEthernet3/2     unassigned      YES unset  up                    up
GigabitEthernet3/3     unassigned      YES unset  up                    up
Port-channel1          10.0.0.1        YES NVRAM  up                    up
ASW1#
```

COMMANDS LEARNED IN THIS CHAPTER
SW(config) port-channel load-balance *mode*
Configures the EtherChannel load-balancing method on a 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 a SWITCH
SW# show etherchannel port-channel
Displays information about the virtual port-channel interfaces on a SWITCH

SwitchA \ SwitchB	off	auto	desirable	passive	active	on
off	NO	NO	NO	NO	NO	NO
auto	NO	NO	PAgP	NO	NO	NO
desirable	NO	PAgP	PAgP	NO	NO	NO
passive	NO	NO	NO	NO	LACP	NO
active	NO	NO	NO	LACP	LACP	NO
on	NO	NO	NO	NO	NO	ON

24. DYNAMIC ROUTING

WHAT IS DYNAMIC ROUTING?

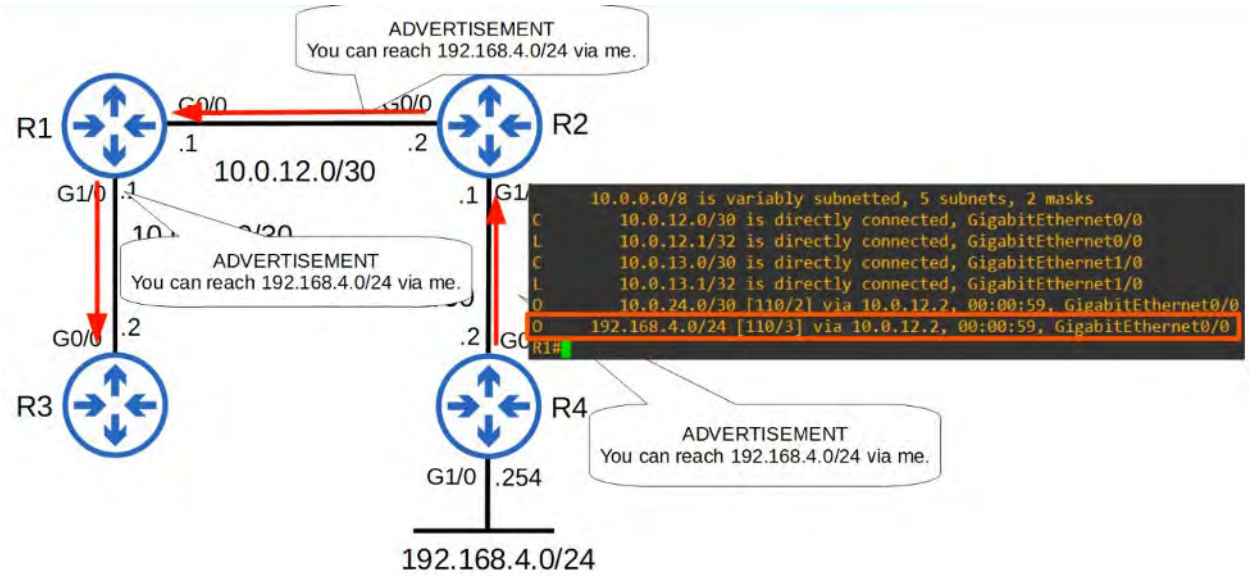


- LAYER 3
- Involves configuring a DYNAMIC ROUTING PROTOCOL on the ROUTER and letting the ROUTER take care of finding the best routes to DESTINATION NETWORKS.
- Not Fixed (will adapt to changes in the LAN)

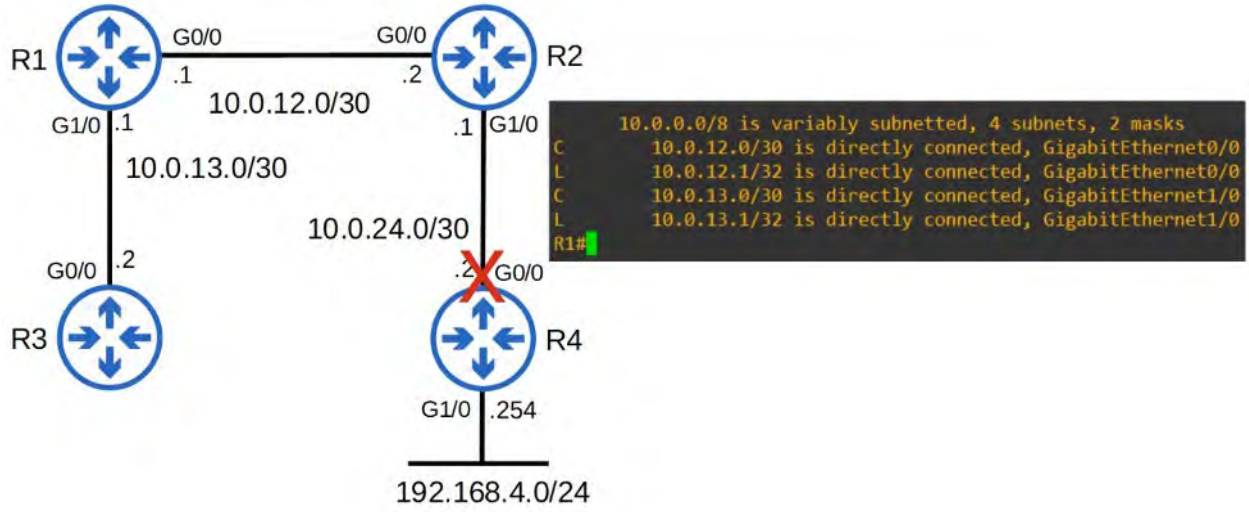
```
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
R1#
```

💡 A NETWORK ROUTE : A ROUTE to a NETWORK or SUBNET (Mask Length < /32)
Ex: **10.0.12.0/30** and **10.0.13.0/30** (above) are NETWORK ROUTES
💡 A HOST ROUTE : A ROUTE to a specific HOST (/32 Mask)
Ex: **10.0.12.1/32** and **10.0.13.1/32** (above) are HOST ROUTES
These two ROUTES were AUTOMATICALLY added to R1's G0/0 and G1/0s INTERFACES

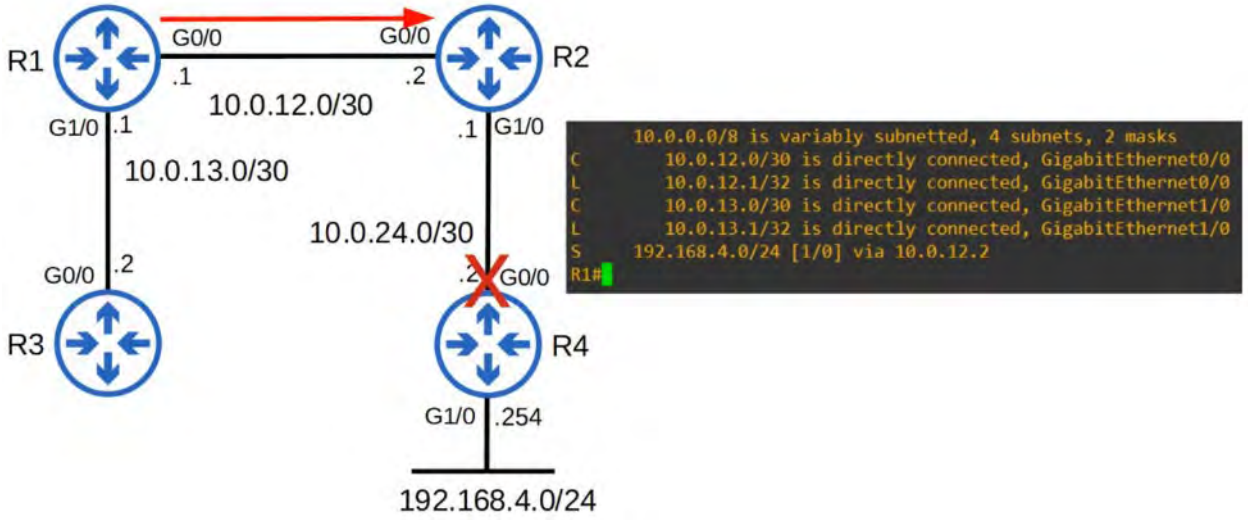
HOW DYNAMIC ROUTING WORKS ?



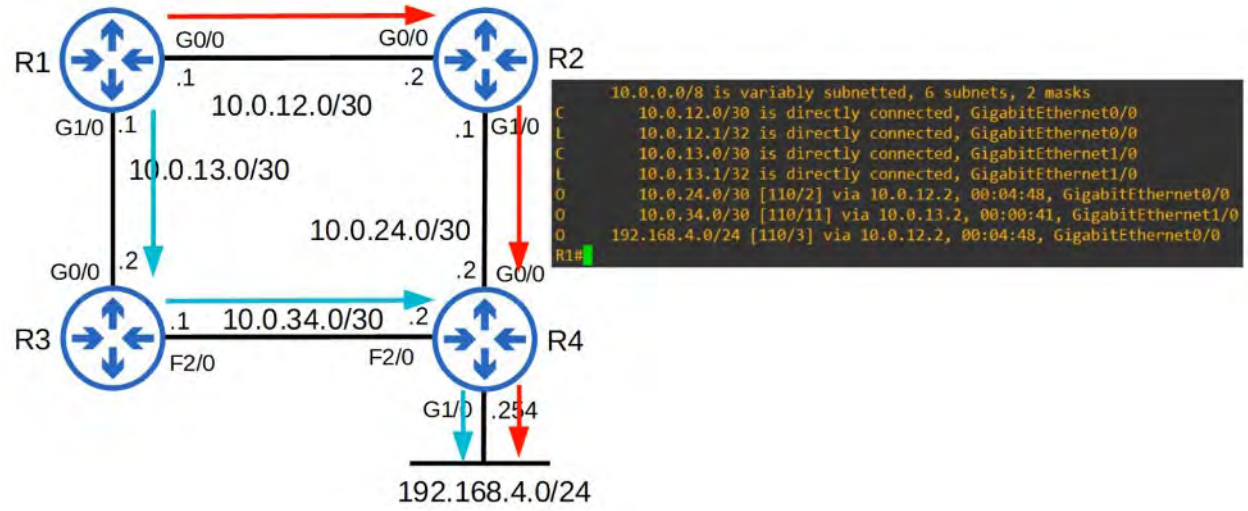
(R4 ADVERTISES to R2 who ADVERTISES to R1 who ADVERTISES to R3 - They add the NETWORK ROUTE to R4 in their ROUTE TABLE)
If the NETWORK ROUTE breaks, the ROUTE is DYNAMICALLY REMOVED from the ROUTE TABLE



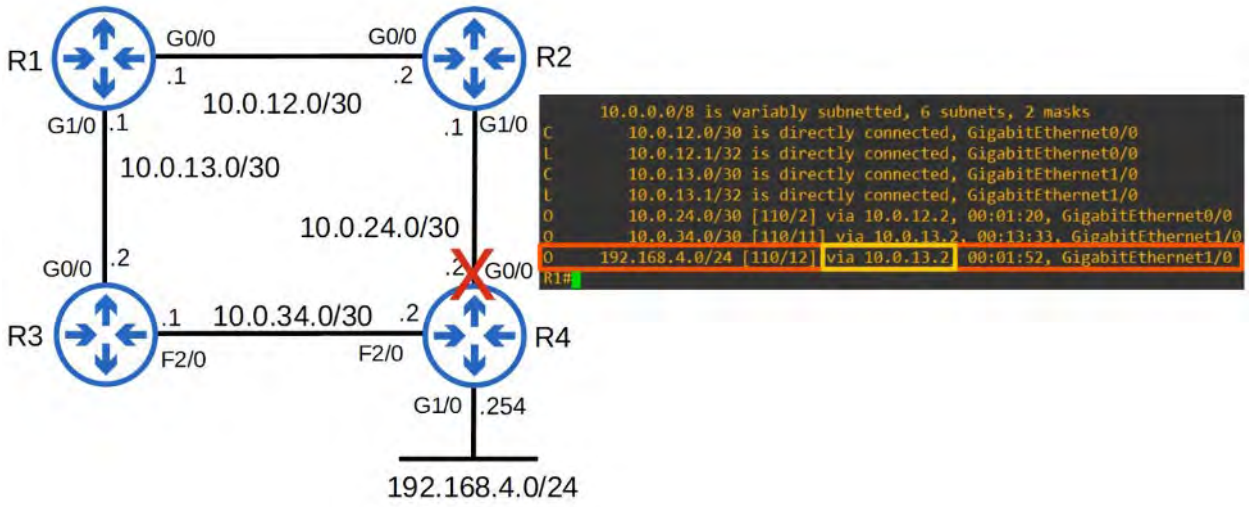
(R1 removing the ROUTE to R4 from it's ROUTE TABLE)
IN STATIC ROUTING, a downed ROUTER will still have traffic passed to it. The ROUTE TABLES are unchanged.



(R1 has a STATIC ROUTE to R4 and passes traffic destined to it's NETWORK regardless of status)
DYNAMIC ROUTING is good but still requires REDUNDANCY so we add another connection between R3 and R4



(Secondary DYNAMIC ROUTE added to R4 from R1 via R3. ROUTE TABLE updated appropriately)
A failure in the ROUTE, via R2 to R4's G0/0 INTERFACE, automatically reroutes traffic via R3



Why does the path prefer using R2's path versus R3?
Because of COST ! This is similar to how SPANNING-TREE works (with SWITCHES)

INTRODUCTION TO DYNAMIC ROUTING PROTOCOLS

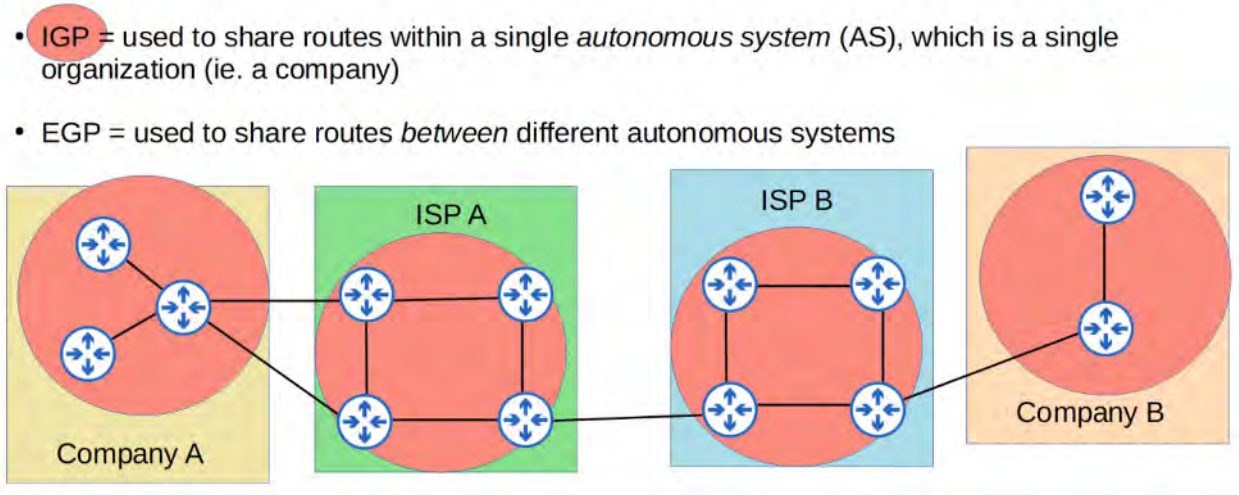
- ROUTERS can use DYNAMIC ROUTING PROTOCOLS to ADVERTISE information about the ROUTES they know to OTHER ROUTES
- They form 'ADJACENCIES' / 'NEIGHBOR RELATIONSHIPS' / 'NEIGHBORSHIPS' with ADJACENT ROUTERS to exchange this information
- If multiple ROUTES to a DESTINATION are learned, the ROUTER determines which ROUTE is SUPERIOR and adds it to the ROUTING TABLE. It uses the 'METRIC' of the ROUTE to decide which is superior (lower metric = superior)

TYPES OF DYNAMIC ROUTING PROTOCOLS
DYNAMIC ROUTING PROTOCOLS can be divided into TWO main categories:

- IGP (Interior Gateway Protocol)
- EGP (Exterior Gateway Protocol)

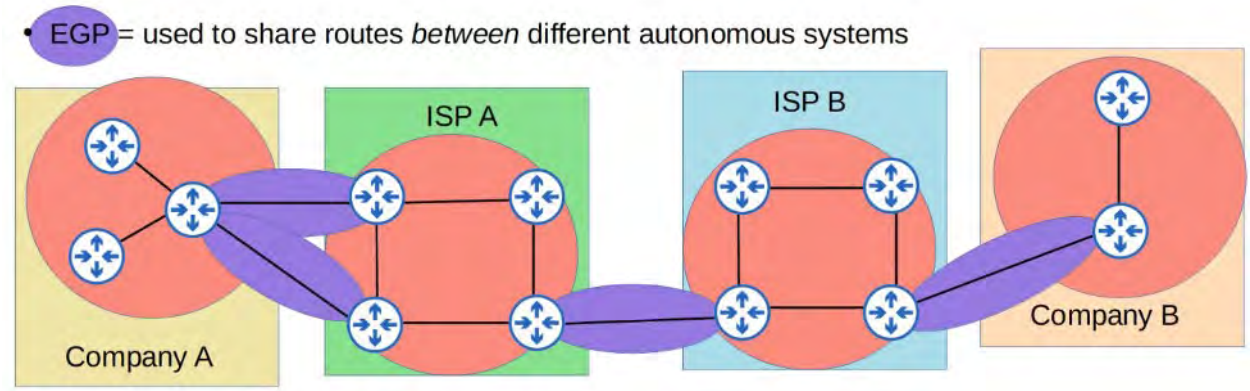
IGP

- Used to SHARE ROUTES within a single *autonomous system* (AS), which is a single organization (ie: a company)

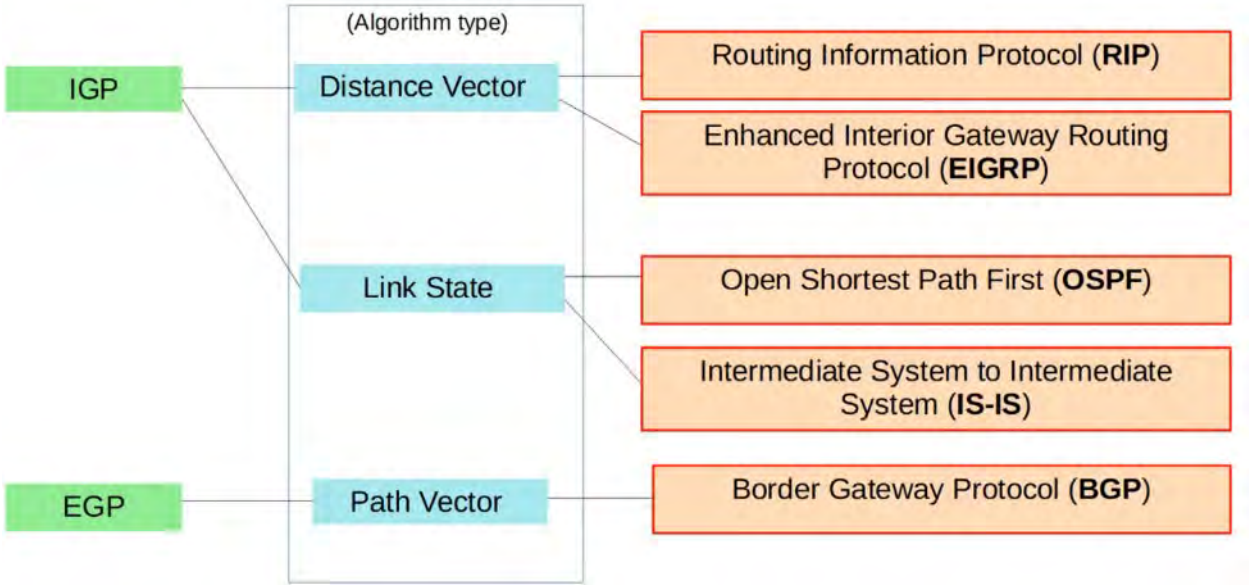


EGP

- Used to SHARE ROUTES *between* different *autonomous systems (AS)*



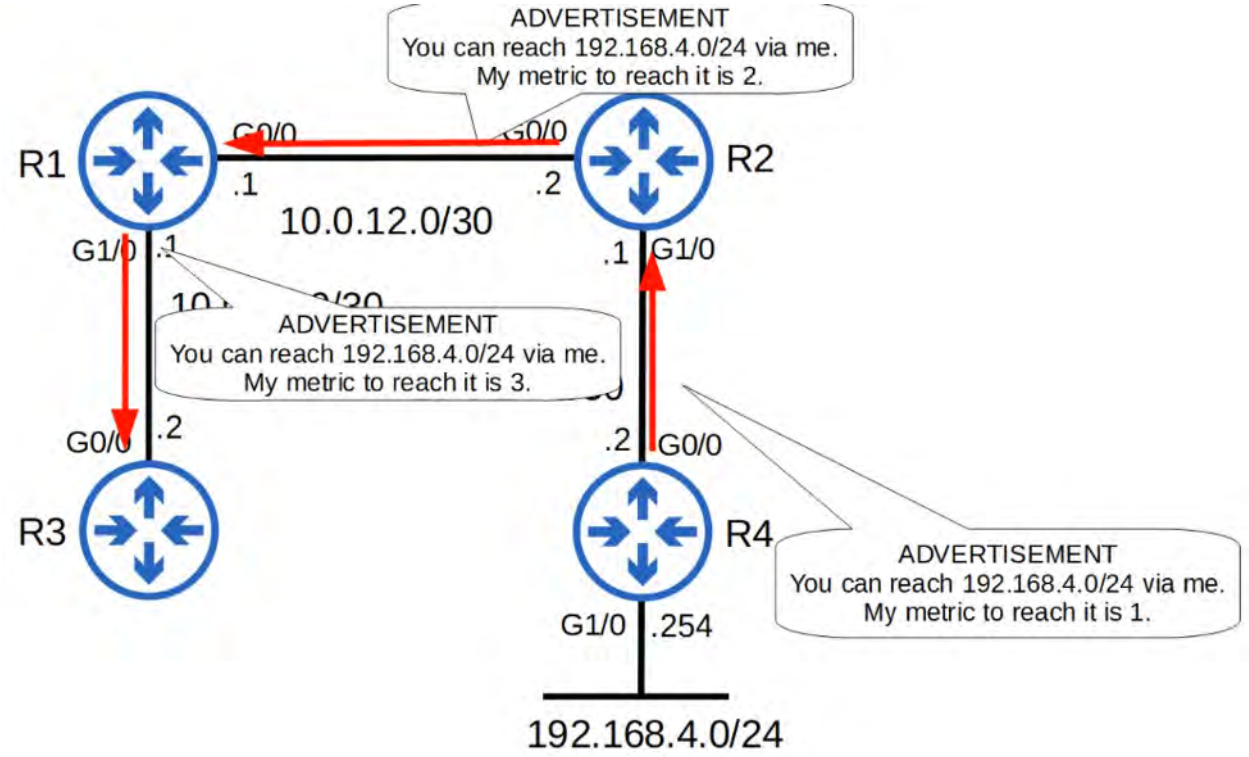
Algorithms used for IGP and EGP and the PROTOCOL for each



💡 YOU MUST MEMORIZE WHICH ALGORITHM IS USED FOR EACH PROTOCOL FOR THE CCNA!

DISTANCE VECTOR ROUTING PROTOCOLS

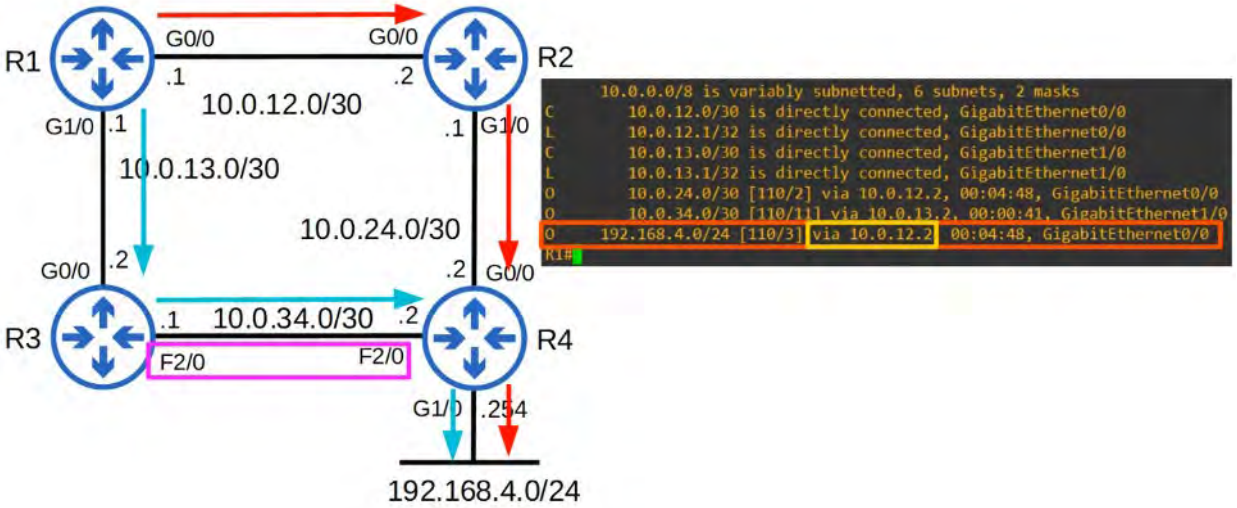
- Called DISTANCE VECTOR because the ROUTERS only learn the 'distance' (METRIC) and 'vector' (DIRECTION, NEXT-HOP ROUTER) of each ROUTE
- DISTANCE VECTOR PROTOCOLS were invented before LINK STATE PROTOCOLS
- Early examples are RIPv1 and Cisco's IGRP (which was updated to EIGRP)
- DISTANCES VECTOR PROTOCOLS operate by sending the following to their directly connection neighbors:
    - Their KNOWN DESTINATION networks
    - Their METRIC to reach their KNOWN DESTINATION networks
- This METHOD of sharing ROUTE information is often called ***'routing by rumor'***
    - ***'routing by rumor'*** = because the ROUTER doesn't know about the NETWORK beyond it's NEIGHBOURS. It only knows the information that the NEIGHBOURS tell it.



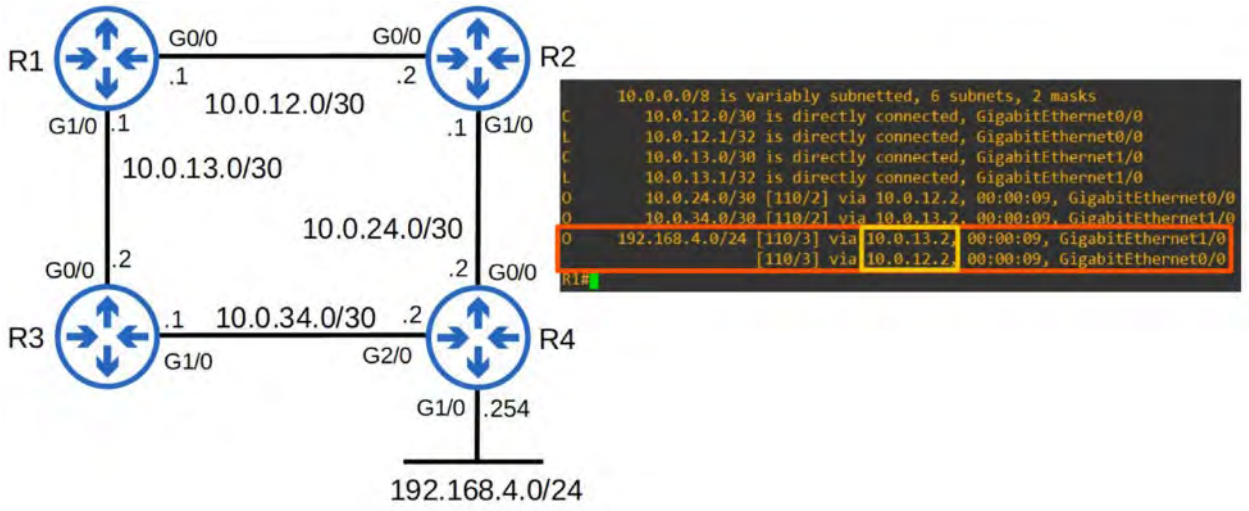
DYNAMIC ROUTING PROTOCOL METRICS

- A ROUTER'S ROUTE TABLE contains the BEST ROUTE to each DESTINATION NETWORK it knows about

If a ROUTER using a DYNAMIC ROUTING PROTOCOL learns TWO different routes to the same DESTINATION, how does it determine which is **'best'** ?
It uses the METRIC value of the ROUTES to determine which is BEST.
A lower METRIC = BETTER! (just like STP)
EACH ROUTING PROTOCOL uses a different METRIC to determine which ROUTE is best



The above choose the RED PATH because the "cost", using R3 F2/0 and R4 F2/0 (FastEthernet) is HIGHER than the R2 G1/0 and R4 G0/0 (GigabyteEthernet)
What if BOTH connections were GigabyteEthernet? (ie: the same METRIC value)



BOTH ROUTES are added to the ROUTE TABLE
So …
💡 If a ROUTER learns TWO (or more) ROUTES via the same ****ROUTING PROTOCOL to the same DESTINATION (same network address, same subnet mask) with the same METRIC, both will be added to the routing table. Traffic will be LOAD-BALANCED over both ROUTES

```
R1#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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
O        10.0.24.0/30 [110/2] via 10.0.12.2, 00:00:09, GigabitEthernet0/0
O        10.0.34.0/30 [110/2] via 10.0.13.2, 00:00:09, GigabitEthernet1/0
O     192.168.4.0/24 [110/3] via 10.0.13.2, 00:00:09, GigabitEthernet1/0
                     [110/3] via 10.0.12.2, 00:00:09, GigabitEthernet0/0
R1#
```

“O” = OSPF PROTOCOL (next to ROUTES)
[110/3] :

- the “3” part is the METRIC.
- the “110” part is ADMINISTRATIVE DISTANCE (covered later)

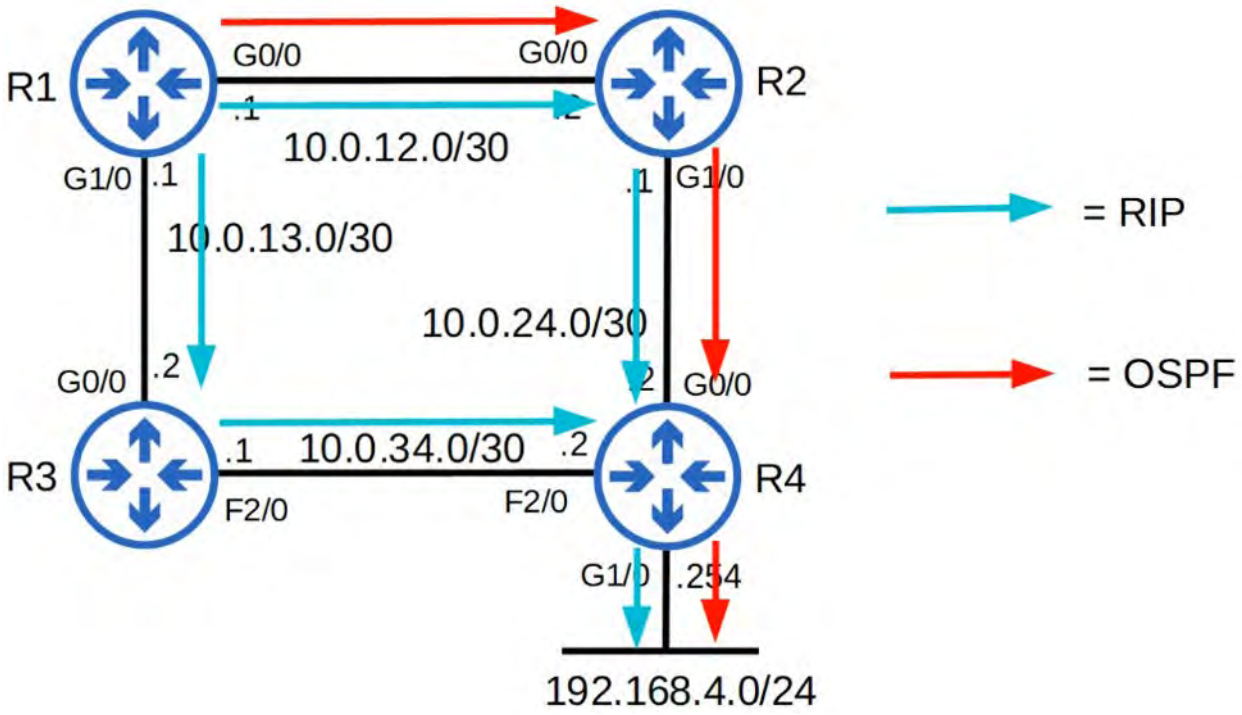
💡 Since BOTH ROUTES share the same METRIC, this is called ECMP (EQUAL COST MULTI-PATH)
You can have ECMP with STATIC ROUTES, as well (they don’t use METRIC, however)

SUMMARY OF DIFFERENT METRICS

Dynamic Routing Protocol Metrics

IGP	Metric	Explanation
RIP	Hop count	Each router in the path counts as one 'hop'. The total metric is the total number of hops to the destination. **Links of all speeds are equal.**
EIGRP	Metric based on bandwidth & delay (by default)	Complex formula that can take into account many values. By default, the bandwidth of the **slowest link in the route** and the total delay of all links in the route are used.
OSPF	Cost	The cost of each link is calculated based on bandwidth. The total metric is the total cost of each link in the route.
IS-IS	Cost	The total metric is the total cost of each link in the route. The cost of each link is **not** automatically calculated by default. All links have a cost of 10 by default.

(IS-IS won't be covered in detail)
EXAMPLE



Using RIP, both ROUTES would be put in R1's ROUTE TABLE
Using OSPF, only the ROUTE from R1 > R2 > R4 would be added to R1's ROUTE TABLE because of the TOTAL COST of each link.
However, BOTH METRICS are trying to achieve the same thing :
To let the ROUTER select the BEST ROUTE to the DESTINATION

ADMINISTRATIVE DISTANCE

- In MOST cases, a company will only use a single IGP - usually OSPF or EIGRP
- However, in some RARE cases, they might use TWO.

    - Ex: If TWO companies connect their networks to share information, TWO different ROUTING PROTOCOLS might be in use.
- METRIC is used to compare ROUTES learned via the same ROUTING PROTOCOL
- Different ROUTING PROTOCOLS use totally different METRICS, so they cannot be compared
    - An OSPF ROUTE to 192.168.4.0/24 might have a METRIC of 30, while an EIGRP ROUTE to the same DESTINATION has a METRIC of 33280. Which ROUTE is better? Which route should the ROUTER put in the ROUTE TABLE ?
- The **ADMINISTRATIVE DISTANCE (AD)**, is used to determine which ROUTING PROTOCOL is preferred.
    - A LOWER AD is preferred, and indicates that the ROUTING PROTOCOL is considered more 'trustworthy' (more likely to select good ROUTES)

ADMINISTRATIVE DISTANCE NUMBERS

Route protocol/type	AD
Directly connected	0
Static	1
External BGP (eBGP)	20
EIGRP	90
IGRP	100
OSPF	110

Route protocol/type	AD
IS-IS	115
RIP	120
EIGRP (external)	170
Internal BGP (iBGP)	200
Unusable route	255

(USE THE FLASHCARDS TO MEMORIZE THESE)

💡 IF the ADMINISTRATIVE DISTANCE is 255, the ROUTER does not believe the SOURCE of that ROUTE and does not install the ROUTE in the ROUTING TABLE!

- The following routes to the destination network 10.1.1.0/24 are learned:
  → next hop 192.168.1.1, learned via RIP, metric 5
  → next hop 192.168.2.1, learned via RIP, metric 3
  → next hop 192.168.3.1, learned via OSPF, metric 10

Which route to 10.1.1.0/24 will be added to the route table?

METRIC is used to COMPARE ROUTES learned from the SAME ROUTING PROTOCOL
However, before comparing METRICS, AD is used to select the BEST ROUTE
Therefore, the BEST ROUTE is :
"next hop 192.168.3.1, learned via OSPF (lower AD than RIP), metric 10"

- You can CHANGE the AD of a ROUTING PROTOCOL (This will be demonstrated in the lecture for OSPF CONFIGURATION)
- You can also change the AD of a STATIC ROUTE:

```
R1(config)#ip route 10.0.0.0 255.0.0.0 10.0.13.2 ?
  <1-255>     Distance metric for this route
  multicast   multicast route
  name        Specify name of the next hop
  permanent   permanent route
  tag         Set tag for this route
  track       Install route depending on tracked item
  <cr>

R1(config)#ip route 10.0.0.0 255.0.0.0 10.0.13.2
```

```
R1(config)#ip route 10.0.0.0 255.0.0.0 10.0.13.2 100
R1(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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
S        10.0.0.0/8 [100/0] via 10.0.13.2
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
D        10.0.24.0/30 [90/3072] via 10.0.12.2, 00:06:35, GigabitEthernet0/0
R1(config)#
```

WHY WOULD YOU WANT TO DO THIS?
FLOATING STATIC ROUTES

- By CHANGING the AD of a STATIC ROUTE, you can make it less preferred than ROUTES learned by a DYNAMIC ROUTING PROTOCOL to the same DESTINATION (make sure the AD is HIGHER than the ROUTING PROTOCOL's AD!)
- This kind of ROUTE is called a 'FLOATING STATIC ROUTE'
- The ROUTE will be inactive (not in the ROUTING TABLE) unless the ROUTE learned by the DYNAMIC ROUTING PROTOCOL is removed.
    - **Ex:** The remote ROUTER stops ADVERTISING it for some reason, or an INTERFACE failure causes an ADJACENCY with a NEIGHBOR to be lost.

LINK STATE ROUTING PROTOCOLS

- When using a LINK STATE ROUTING PROTOCOL, every ROUTER creates a 'connectivity map' of the NETWORK
- To allow this, each ROUTER ADVERTISES information about its INTERFACES (connected NETWORKS) to its NEIGHBOURS. These ADVERTISEMENTS are passed along to the other ROUTERS, until all ROUTERS in the NETWORK develop the same map of the NETWORK
- Each ROUTER independently uses this MAP to calculate the BEST ROUTES to each DESTINATION

- LINK STATE PROTOCOLS use more resources (CPU) on the ROUTER, because MORE information is shared.
- However, LINK STATE PROTOCOLS tend to be FASTER in reacting to CHANGES in the NETWORK than DISTANCES VECTOR PROTOCOLS

25. RIP and EIGRP (IGP : DYNAMIC VECTOR)

ROUTING INFORMATION PROTOCOL (RIP)

- Routing Information Protocol (Industry Standard)
- is a DISTANCE VECTOR IGP
    - uses Routing-By-Rumor logic to learn/share routes
- Uses HOP COUNT as it's METRIC (One Router = One Hop) Bandwidth is irrelevant
- MAX HOP COUNT is 15 (anything more 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 neighbour ROUTERS to send their ROUTING TABLE
    - RESPONSE:
        - To SEND the LOCAL router's ROUTING TABLE to neighbouring ROUTERS

By DEFAULT, RIP-Enabled ROUTERS will share their ROUTING TABLE every 30 seconds
RIPv1 and RIPv2
RIPv1:

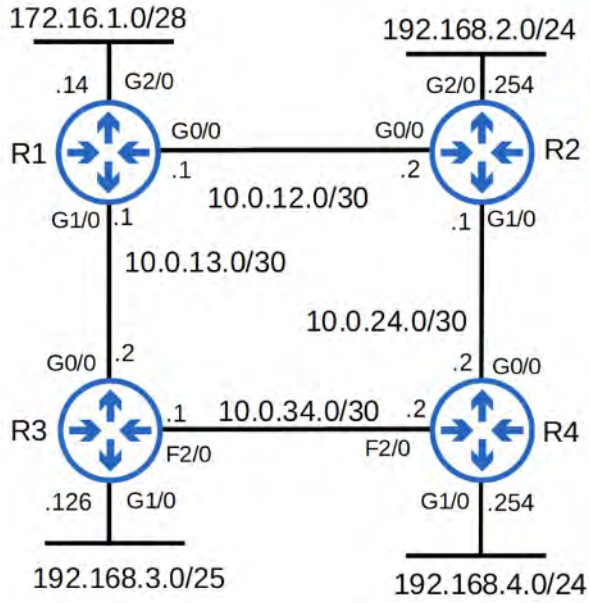
- Only advertises *classful addresses* (Class A, Class B, Class C)
- Doesn't support VLSM, CIDR
- Doesn't include SUBNET MASK information in ADVERTISEMENTS (RESPONSE messages)
    - Example:
        - 10.1.1.0/24 will become 10.0.0.0 (Class A Address, so assumed to be /8)
        - 172.16.192.0/18 will become 172.16.0.0 (Class B Address, so assumed to be /16)
        - 192.168.1.40/30 will become 172.168.1.0 (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 to have joined that specific ***multicast group***

CONFIGURING RIP

RIP Configuration




```
R1(config)#router rip
R1(config-router)#version 2
R1(config-router)#no auto-summary
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.0.0
```

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

- Look for INTERFACES with an IP ADDRESS that is in the specific RANGE
- ACTIVATES 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

Because the RIP "network" command is CLASSFUL. It will automatically convert to CLASSFUL networks

- 10.0.0.0 is assumed to be 10.0.0.0/8
- R1 will look for ANY INTERFACES with an IP ADDRESS that matches 10.0.0.0/8 (because it is /8 it only needs to match the FIRST 8 bits)
- 10.0.12.1 and 10.0.13.1 both match SO RIP is ACTIVATED on G0/0 and G0/1
- R1 then forms ADJACENCIES with its neighbors R2 and R3
- R1 ADVERTISES 10.0.12.0/30 and 10.0.13.0/30 (NOT 10.0.0.0/8) to it's RIP neighbors

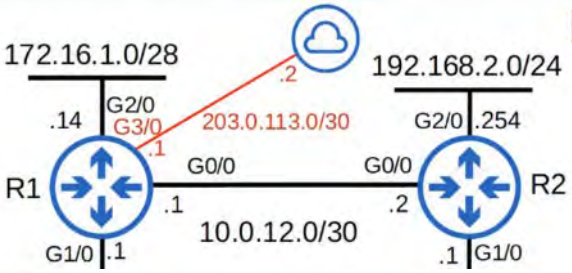
```
R1(config-router)#network 172.16.0.0
```

- Because the "network" command is CLASSFUL, 172.16.0.0 is assumed to be 172.16.0.0/16
- R1 will look for ANY INTERFACES that match 172.16.0.0/16
- 172.16.1.14 matches, so R1 will ACTIVATE RIP on G2/0
- There are NO RIP neighbors connected to G2/0 so no NEW ADJACENCIES are formed
    - Although there are NO RIP neighbors, R1 will still send ADVERTISEMENTS out of G2/0.
    - This is unnecessary traffic, so G2/0 should be configured as a **passive interface**

```
R1(config-router)#passive-interface g2/0
```

- the "passive-interface" command tells the ROUTER to stop sending RIP advertisements out of the specified interface (G2/0)
- However, the ROUTER will continue to ADVERTISE the network prefix of the interface (172.16.1.0/28) to it's RIP neighbors (R2, R3)
- You should ALWAYS use this command on INTERFACES which don't have any RIP neighbors
- EIGRP and OSPF both have the same passive INTERFACE functionality, using the same command.

HOW TO ADVERTISE A DEFAULT ROUTE INTO RIP




```
R1(config)#ip route 0.0.0.0 0.0.0.0 203.0.113.2
```

```
Gateway of last resort is 203.0.113.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 203.0.113.2
```

To SHARE this DEFAULT ROUTE with R1's RIP neighbors, using this command:

```
R1(config-router)#default-information originate
```

RIP doesn't care about interface AD cost (RIP cost is 120), only "hops".
Since both have an equal number of "hops", both paths appear in the DEFAULT ROUTE (Gateway of Last Resort)

```
Gateway of last resort is 10.0.34.1 to network 0.0.0.0

R*    0.0.0.0/0 [120/2] via 10.0.34.1, 00:00:06, FastEthernet2/0
                [120/2] via 10.0.24.1, 00:00:01, GigabitEthernet0/0
```

"show ip protocols" (for RIP)

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

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 28 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0    2     2
    GigabitEthernet1/0    2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.0.0
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.0.12.2            120      00:00:21
    10.0.13.2            120      00:00:06
  Distance: (default is 120)
```

"Maximum path: 4" is the DEFAULT but can be changed with this command:

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

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

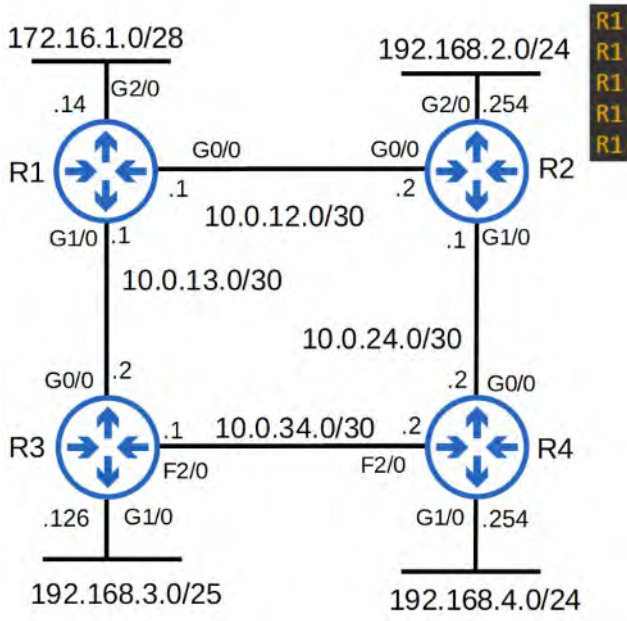
“Distance” (AD) can be changed with this command (DEFAULT is 120)

```
R1(config-router)#distance ?
  <1-255>  Administrative distance

R1(config-router)#distance 85
```

ENHANCED INTERIOR GATEWAY ROUTING PROTOCOL (EIGRP)

- Enhanced Interior Gateway Routing Protocol
- is a DISTANCE VECTOR IGP
- Was Cisco proprietary, but Cisco has now published it openly so other vendor 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 (Memorize this number)**
- Is the ONLY IGP that can perform **unequal**-cost load-balancing (by DEFAULT, it performs ECMP load-balancing over 4 paths like RIP)

CONFIGURATION OF EIGRP




```
R1(config)#router eigrp 1
R1(config-router)#no auto-summary
R1(config-router)#passive-interface g2/0
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.1.0 0.0.0.15
```

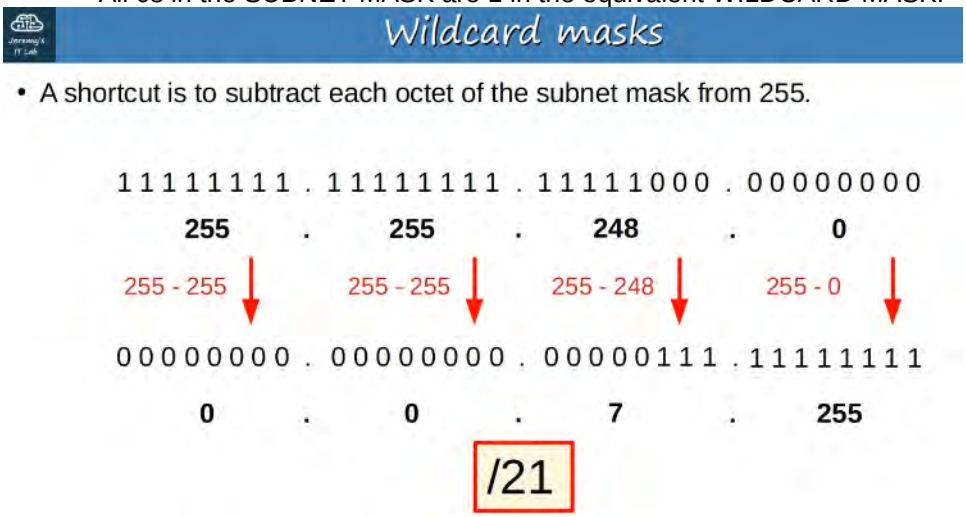
“router eigrp ”

- The AS (Autonomous System) number MUST MATCH between ROUTERS or they will NOT form an ADJACENCY and share ROUTE information
- Auto-summary might be ENABLED or DISABLED by DEFAULT; depending on the ROUTER/IOS version. If ENABLED, DISABLE it.

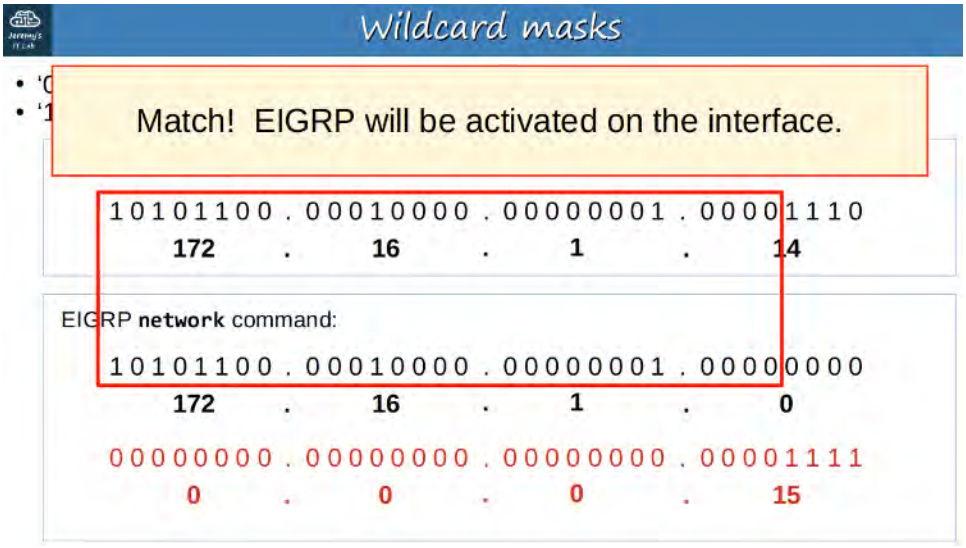
- The **"network"** command will assume a CLASSFUL ADDRESS, if you don't specify the SUBNET MASK
- EIGRP uses a *wildcard mask* instead of a regular subnet mask

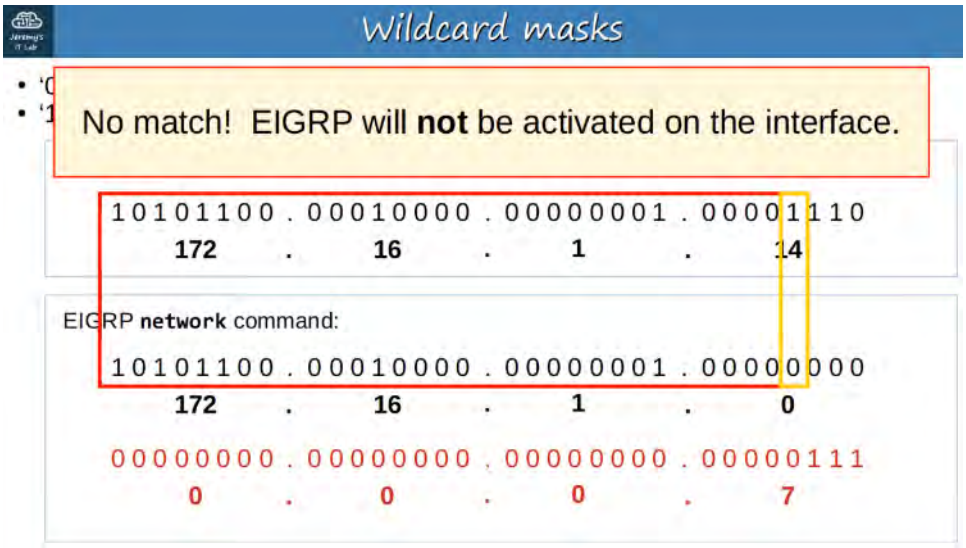
A WILDCARD MASK is an "inverted" SUBNET MASK

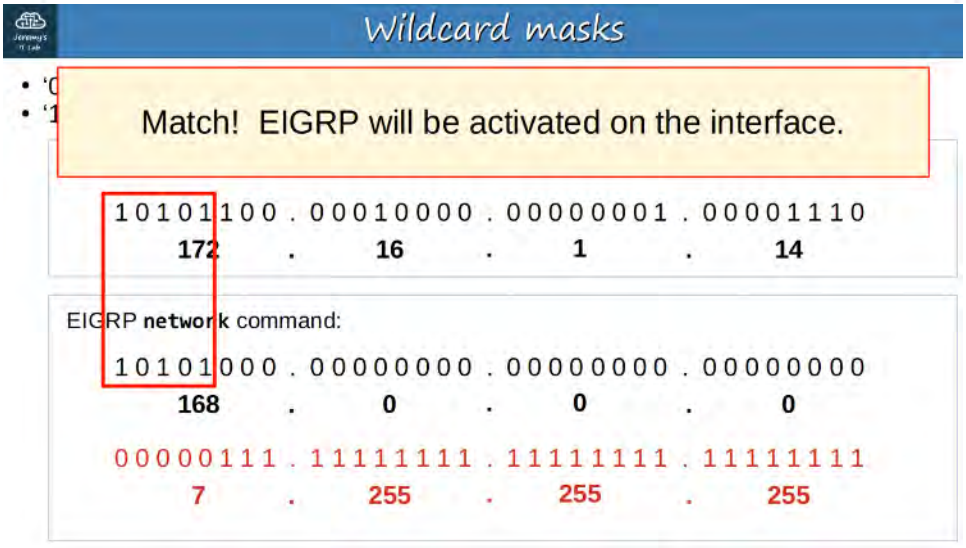
- All 1's in the SUBNET MASK are 0 in the equivalent WILDCARD MASK.
- All 0s in the SUBNET MASK are 1 in the equivalent WILDCARD MASK.



"0" in the WILDCARD MASK = BITS MUST MATCH !
"1" in the WILDCARD MASK = Do not have to match







"show ip protocols" (for EIGRP)

```
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”
ROUTER ID order of priority:

- Manual configuration
- Highest IP ADDRESS on a LOOPBACK INTERFACE
- 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
```

“Distance” (AD)
EIGRP has TWO VALUES:

- Internal = 90
- External = 170

MEMORIZE THESE VALUES!
“show ip route” (for EIGRP)

```
R1#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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
D        10.0.24.0/30 [90/3072] via 10.0.12.2, 00:11:09, GigabitEthernet0/0
D        10.0.34.0/30 [90/28416] via 10.0.13.2, 00:11:09, GigabitEthernet1/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/28 is directly connected, GigabitEthernet2/0
L        172.16.1.14/32 is directly connected, GigabitEthernet2/0
D     192.168.2.0/24 [90/3072] via 10.0.12.2, 00:11:09, GigabitEthernet0/0
      192.168.3.0/25 is subnetted, 1 subnets
D        192.168.3.0 [90/3072] via 10.0.13.2, 00:11:10, GigabitEthernet1/0
D     192.168.4.0/24 [90/3328] via 10.0.12.2, 00:11:09, GigabitEthernet0/0
```

NOTE the large METRIC numbers. This is a DOWNSIDE to EIGRP - even on small networks!

EIGRP METRIC

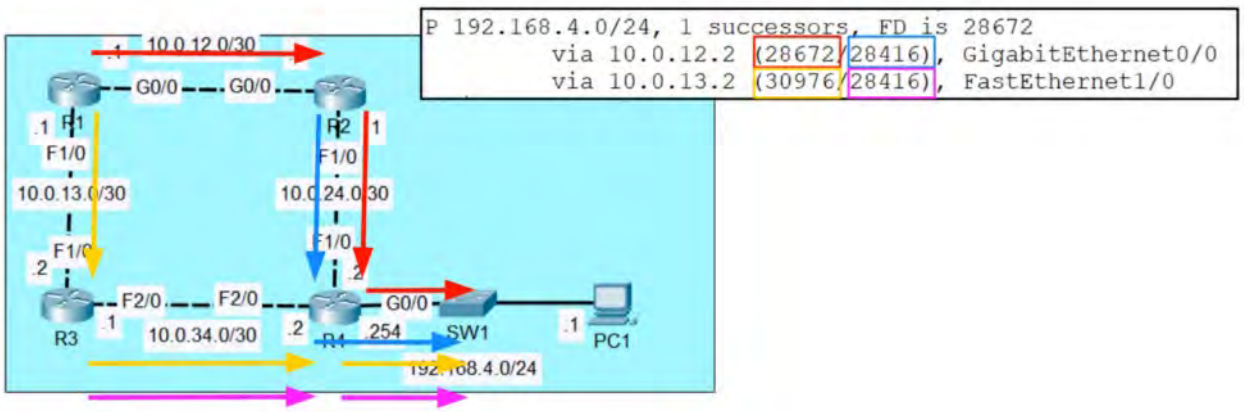
- By DEFAULT, EIGRP uses BANDWIDTH and DELAY to calculate METRIC
- Default “K” values are:
  - K1 = 1, K2 = 0, K3 = 1, K4 = 0, K5 = 0

💡 Simplified calculation : METRIC = BANDWIDTH (Slowest Link) + DELAY (of ALL LINKS)

EIGRP TERMINOLOGY

- **Feasible Distance** = This ROUTER’s METRIC value to the ROUTE’s DESTINATION
- **Reported Distance** (aka **Advertised Distance**) = The neighbor’s METRIC value to the ROUTE’s DESTINATION

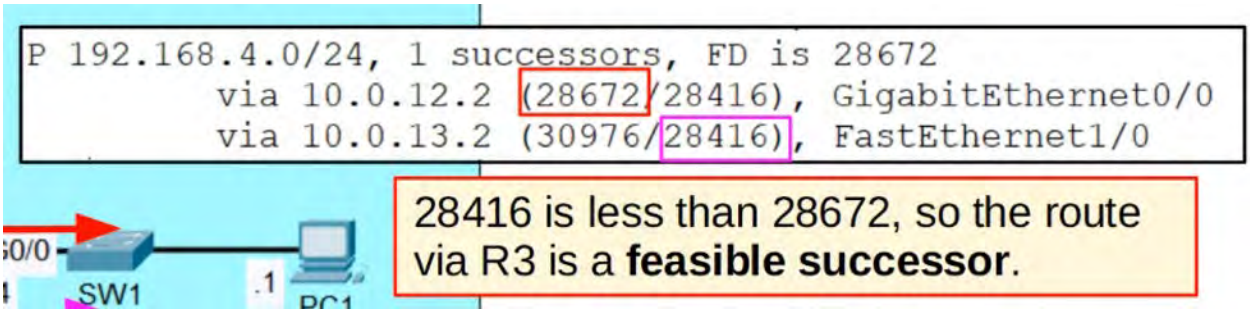
- **Feasible Distance** = This router’s metric value to the route’s destination.
- **Reported Distance** (aka Advertised Distance) = The neighbor’s metric value to the route’s destination.



- **Successor =** the ROUTE with the LOWEST METRIC to the DESTINATION (the best route)

- **Feasible Successor** = An alternate ROUTE to the DESTINATION (not the best route) which meets the *feasibility condition*

FEASIBILITY CONDITION : A ROUTE is considered a ***Feasible Successor*** if it's ***Reported Distance*** is LOWER than the Successor ROUTE's ***Feasible distance***



EIGRP : UNEQUAL-COST LOAD-BALANCED

```
R1#show ip protocols

Routing Protocol is "eigrp  100 "
  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 metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
```

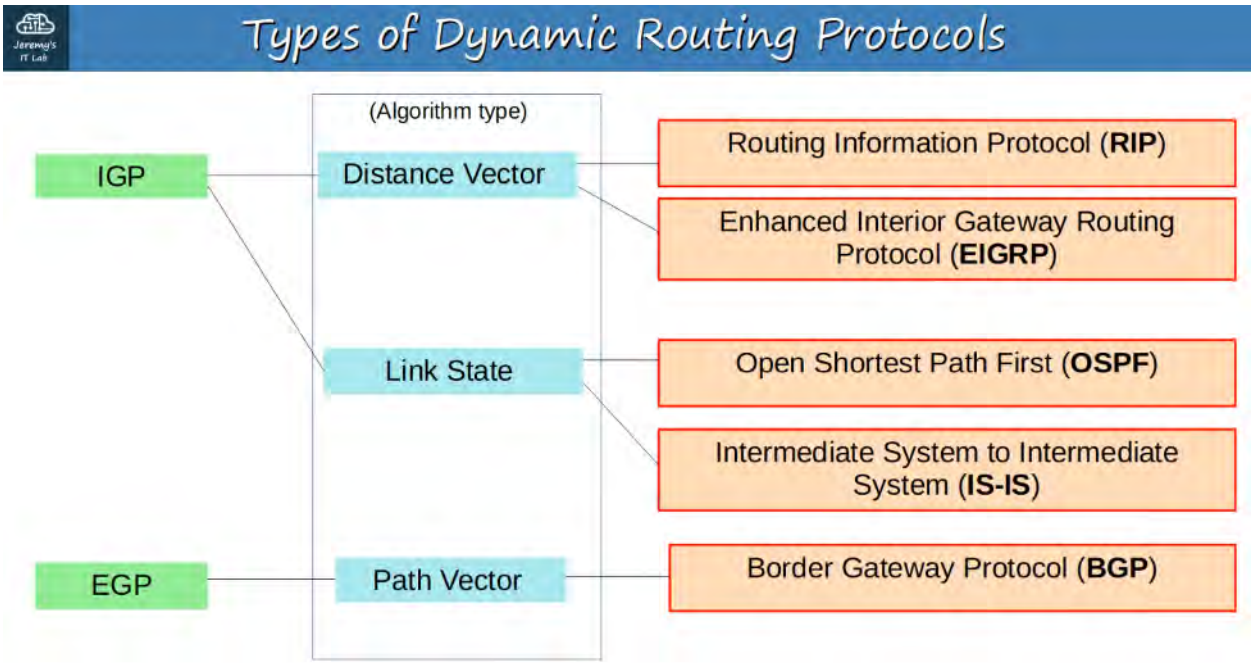
"maximum metric variance 1" = the DEFAULT value
Variance 1 = only ECMP (Equal-Cost Multiple Path) load-balancing will be performed

```
R1(config-router)#variance ?
  <1-128>  Metric variance Multiplier
R1(config-router)#variance 2
```

Variance 2 = ***feasible successor*** routes with an FD up to 2x the ***successor*** route's FD can be used to load-balance

💡 EIGRP will only perform UNEQUAL-COST LOAD-BALANCING over ***feasible successor*** ROUTES. If a ROUTE doesn't meet the ***feasibility condition***, it will NEVER be selected for load-balancing, regardless of **variance**

26. OSPF : PART 1 (IGP : LINK STATE)



LINK STATE ROUTING PROTOCOLS

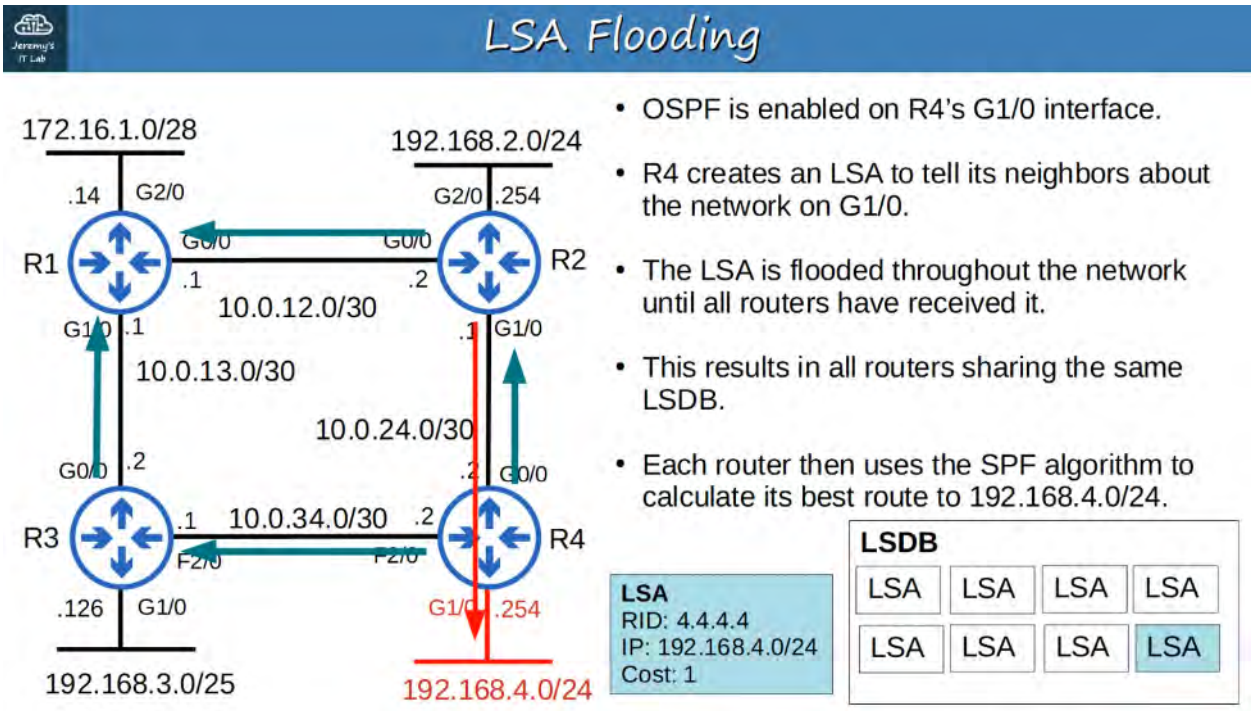
- When using a LINK STATE ROUTING PROTOCOL, every ROUTER creates a 'connectivity map' of the NETWORK
- To allow this, each ROUTER ADVERTISES information about its INTERFACES (connected NETWORKS) to its NEIGHBOURS. These ADVERTISEMENTS are passed along to the other ROUTERS, until all ROUTERS in the NETWORK develop the same map of the NETWORK
- Each ROUTER independently uses this MAP to calculate the BEST ROUTES to each DESTINATION
- LINK STATE PROTOCOLS use more resources (CPU) on the ROUTER, because MORE information is shared.
- However, LINK STATE PROTOCOLS tend to be FASTER in reacting to CHANGES in the NETWORK than DISTANCES VECTOR PROTOCOLS

BASIC OSPF OPERATIONS

- Stands for **Open Shortest Path First**
- Uses the **Shortest Path First** algorithm
    - Created by Dutch comp. scientist - Edsger Dijkstra
    - aka **Dijkstra's Algorithm** (Could be Exam Question)

THREE Versions:

- OSPFv1 (1989) : OLD, not in use anymore
- OSPFv2 (1998) : Used for IPv4
- OSPFv3 (2008) : Used for IPv6 (can be used for IPv4, but v2 is usually 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)



💡 LSA's have an AGING TIMER of 30 Minutes, by Default). The LSA will be FLOODED again after the timer expires

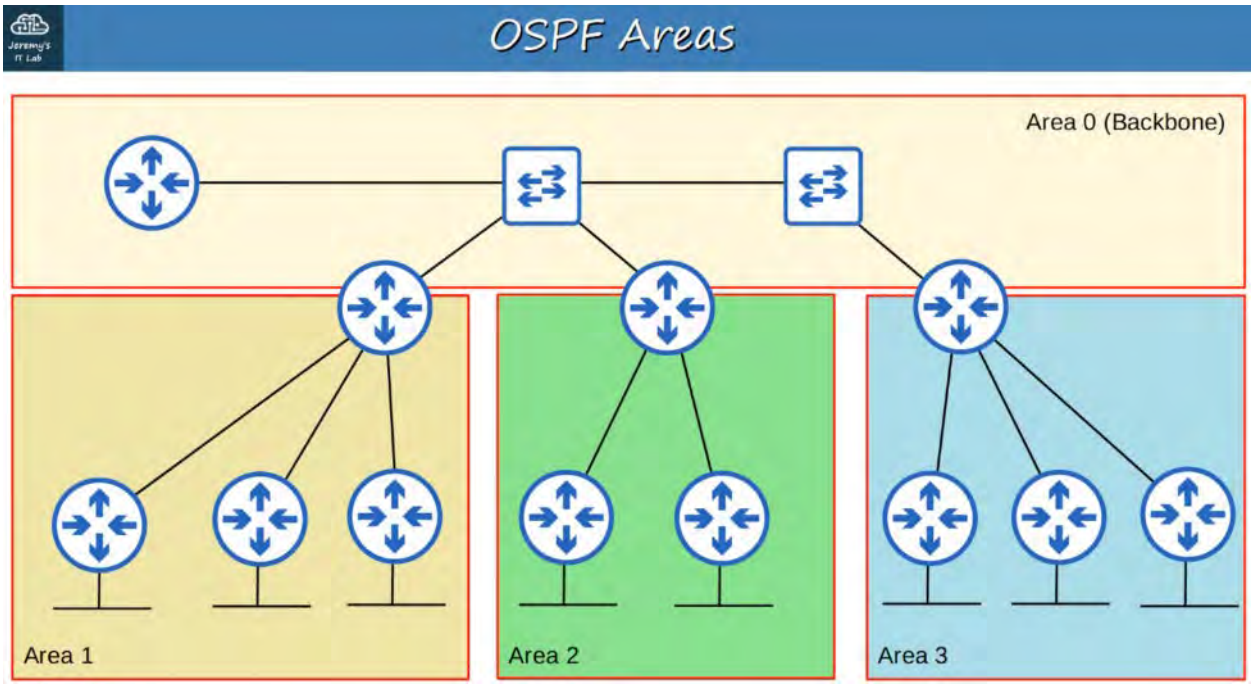
In OSPF, there are THREE MAIN STEPS in the process of sharing LSAs and determining the BEST ROUTE to each DESTINATION in the network

1. **BECOME NEIGHBORS** with other ROUTERS connected to 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
- LARGE NETWORKS, *single-area* design can have NEGATIVE effects:
  - SPF ALGORITHM takes more time to calculate ROUTES
  - SPF ALGORITHM requires exponentially more processing power on ROUTERS
  - Larger LSDB takes up more MEMORY on ROUTERS
  - Small changes in NETWORK cause every ROUTER to FLOOD LSAs and run the SPF algorithm again
- By dividing up a large OSPF NETWORK into several SMALLER ***areas***, you can avoid the above NEGATIVE effects (sounds similar to VLANs re: broadcast domains)

WHAT IS AN OSPF AREA?



- 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)

💡 ABRs maintain a SEPARATE LSDB for each AREA they are connected to.

💡 It is recommended that you connect an ABR to a MAXIMUM of TWO AREAS.

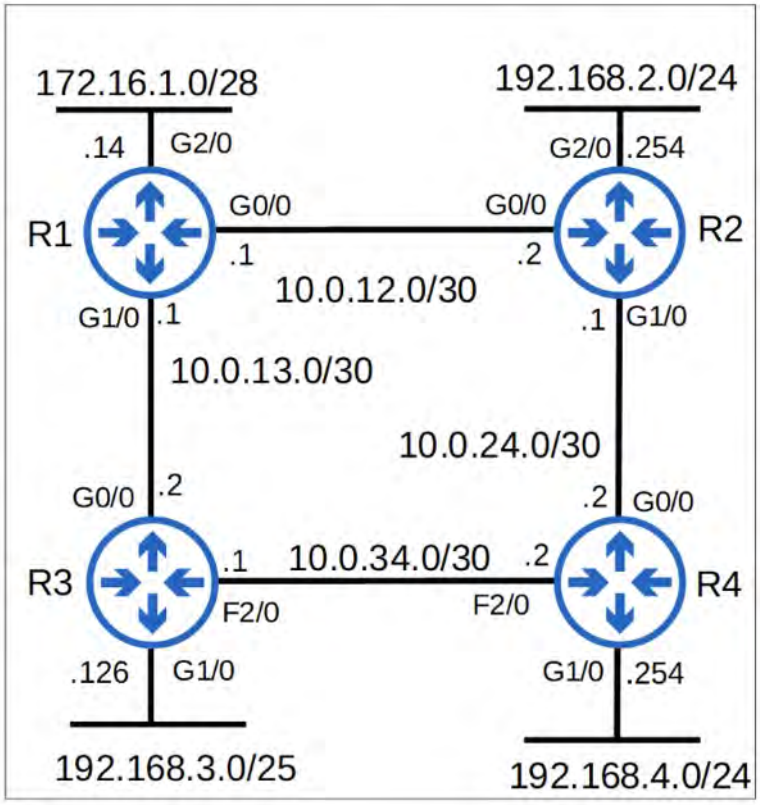
💡 Connecting an ABR to 3+ AREAS can overburden the ROUTER

- 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 INTER-AREA ROUTE is a ROUTE to a DESTINATION in a DIFFERENT OSPF AREA

OSPF RULES

- OSPF AREAS should be CONTIGUOUS (no split AREAS)
- All OSPF AREAS must have *at least* ONE ABR connected to the BACKBONE AREA
- OSPF INTERFACES in the SAME SUBNET *must* be in the SAME AREA

BASIC OSPF CONFIGURATION
OSPF AREA 0



Commands for configuring an OSPF

```
R1(config)#router ospf ?
  <1-65535>  Process ID

R1(config)#router ospf 1
R1(config-router)#network 10.0.12.0 0.0.0.3
% Incomplete command.

R1(config-router)#network 10.0.12.0 0.0.0.3 area 0
R1(config-router)#network 10.0.13.0 0.0.0.3 area 0
R1(config-router)#network 172.16.1.0 0.0.0.15 area 0
R1(config-router)#
```

- The OSPF **Process ID** is **locally significant.** ROUTERS with different Process IDs can become OSPF Neighbors
- The OSPF "network" command requires you to specify the AREA (in this case, it's "area 0")
- For the CCNA, you only need to configure single-area OSPF (AREA 0)

The "network" command tells OSPF to:

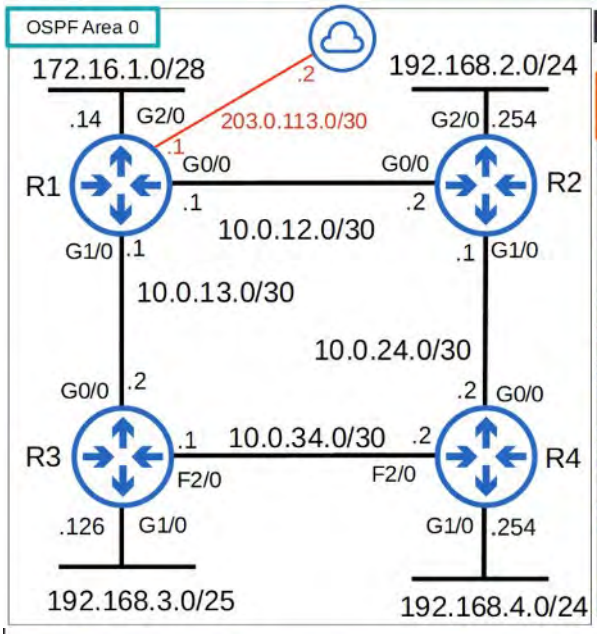
- Look for ANY INTERFACES with an IP ADDRESS contained in the RANGE specified in the "network" command
- Activate OSPF on the INTERFACE in the specified AREA
- The ROUTER will then try to become OSPF neighbors with other OSPF-Activated neighbor ROUTERS

```
R1(config-router)#passive-interface g2/0
```

- Know this command from RIP and EIGRP

- The “passive-interface” command tells the ROUTERS to stop sending OSFP ‘hello’ messages out of the INTERFACE
- However, the ROUTER will continue to send LSA’s informing it’s neighbors about the SUBNET configured on the INTERFACE
- You should ALWAYS USE this command on neighbors which don’t have any OSPF neighbors

Advertise a default route into OSPF




```
R1(config)#ip route 0.0.0.0 0.0.0.0 203.0.113.2
```

```
Gateway of last resort is 203.0.113.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 203.0.113.2
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/2] via 10.0.12.2, 01:22:21, GigabitEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/2] via 10.0.13.2, 01:22:11, GigabitEthernet1/0
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/3] via 10.0.13.2, 00:00:06, GigabitEthernet1/0
                 [110/3] via 10.0.12.2, 00:00:06, GigabitEthernet0/0
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
O        10.0.24.0/30 [110/2] via 10.0.12.2, 01:22:21, GigabitEthernet0/0
O        10.0.34.0/30 [110/2] via 10.0.13.2, 01:22:11, GigabitEthernet1/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/28 is directly connected, GigabitEthernet2/0
L        172.16.1.14/32 is directly connected, GigabitEthernet2/0
O     192.168.2.0/24 [110/2] via 10.0.12.2, 01:22:21, GigabitEthernet0/0
      192.168.3.0/25 is subnetted, 1 subnets
O        192.168.3.0 [110/2] via 10.0.13.2, 01:22:11, GigabitEthernet1/0
O     192.168.4.0/24 [110/3] via 10.0.13.2, 01:22:11, GigabitEthernet1/0
                     [110/3] via 10.0.12.2, 01:22:21, GigabitEthernet0/0
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C        203.0.113.0/30 is directly connected, GigabitEthernet3/0
L        203.0.113.1/32 is directly connected, GigabitEthernet3/0
```

```
R1(config-router)#default-information originate
```

```
R2#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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 10.0.12.1 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/1] via 10.0.12.1, 00:01:38, GigabitEthernet0/0
```

“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
```

```
R1#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
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 1.1.1.1
```

NOTE the "no" in square brackets - this indicates this is the DEFAULT choice

```
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
```

DISTANCE (AD) for OSPF is 110 (DEFAULT) but can be changed with the “distance” command

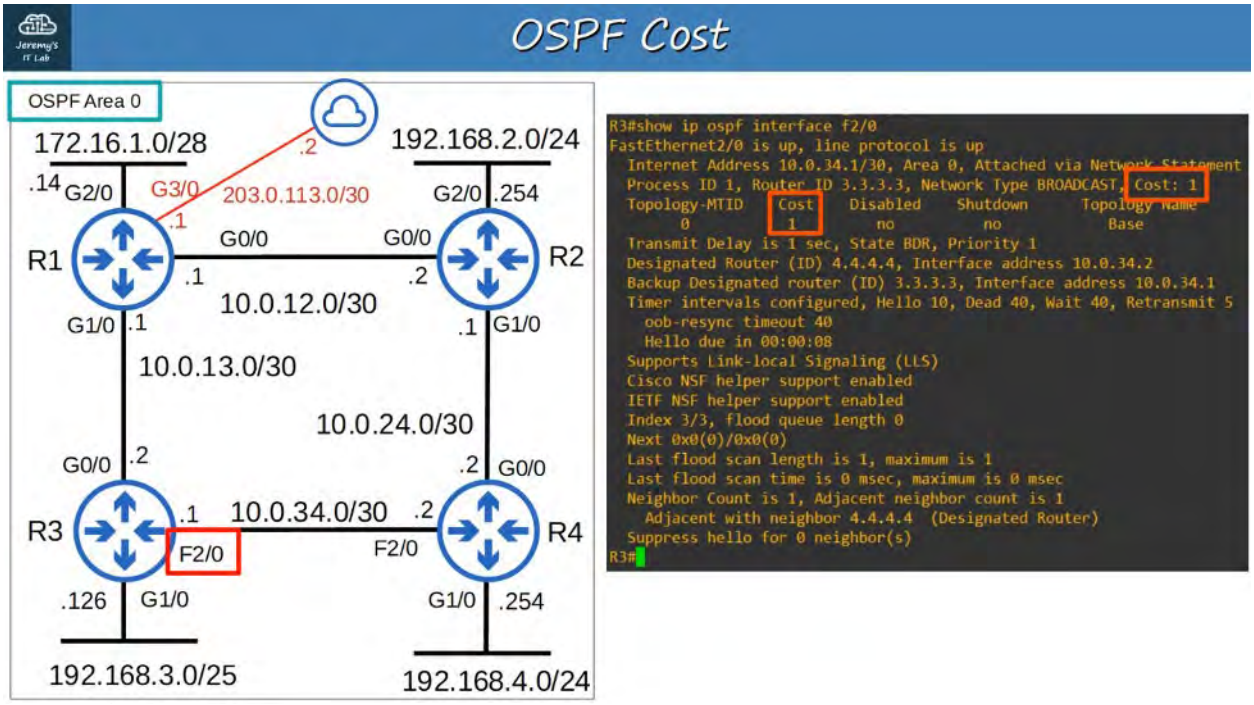
```
R1(config-router)#distance ?
  <1-255>  Administrative distance

R1(config-router)#distance 85
```

27. OSPF : PART 2 (IGP : LINK STATE)

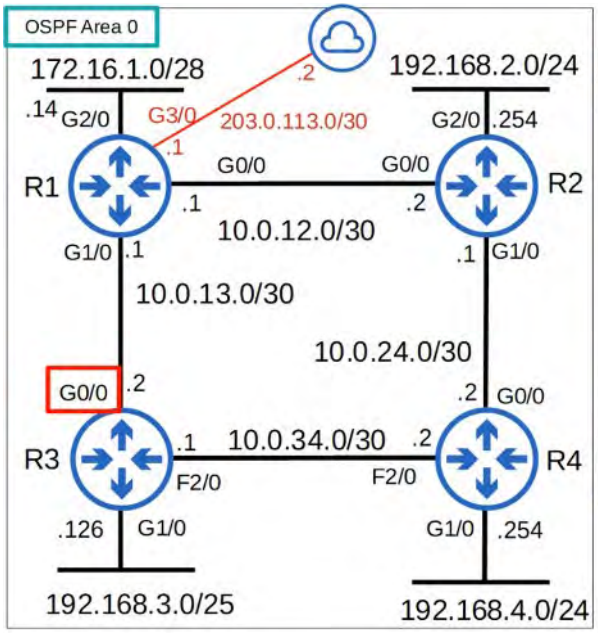
OSPF METRIC (Cost)

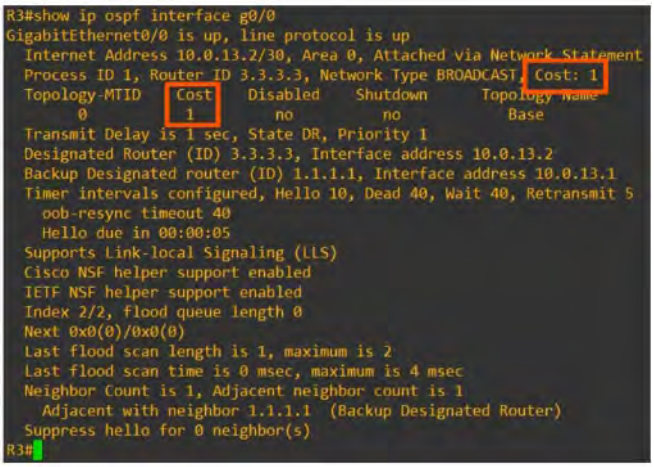
- OSPFs METRIC is called **COST**
- It is automatically calculated based on the bandwidth (SPEED) of the INTERFACE
- It is calculated by DIVIDING a REFERENCE BANDWIDTH value by the INTERFACE bandwidth
- The DEFAULT REFERENCE BANDWIDTH is 100 mbps
    - REFERENCE: 100 mbps / INTERFACE: 10 mbps = COST (10)
    - REFERENCE: 100 mbps / INTERFACE: 100 mbps = COST (1)
    - REFERENCE: 100 mbps / INTERFACE: 1000 mbps = COST (1)
    - REFERENCE: 100 mbps / INTERFACE: 10000 mbps = COST (1)
- ALL COST values less than 1 will be CONVERTED to 1
- Therefore FastEthernet (100 mbps), Gigabit Ethernet (1000 mbps), 10 Gig Ethernet, etc. are EQUAL and all have a COST of 1

FastEthernet COST



Gigabit Ethernet COST





You can (and SHOULD) change the REFERENCE BANDWIDTH with this command:

💡 R1(config-router)# **auto-cost reference-bandwidth** *megabits-per-second*

The command is entered in “megabits per second” (DEFAULT is “100”)

Example: using a value of “100000”

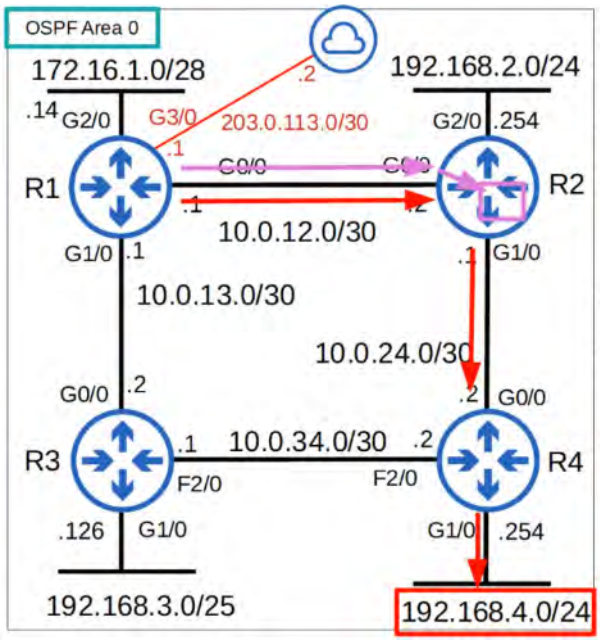
- 100000 / 100 = COST of 1000 for FastEthernet
- 100000 / 1000 = COST of 100 for Gig Ethernet

You should configure a reference bandwidth GREATER than the FASTEST links in your NETWORK (to allow for future upgrades)

Changing the REFERENCE BANDWIDTH needs to be done on ALL OSPF ROUTERS in the NETWORK

THE OSPF COST to a DESTINATION is the TOTAL COST of the ‘outgoing/exit INTERFACES’

LOOPBACK INTERFACES have a COST of 1



- The OSPF cost to a destination is the total cost of the ‘outgoing/exit interfaces’
- For example, R1’s cost to reach 192.168.4.0/24 is :
  100 (R1 G0/0) + 100 (R2 G1/0) + 100 (R4 G1/0) = 300
- Loopback interfaces have a cost of 1
- What is R1’s cost to reach 2.2.2.2 (R2’s loopback0 interface)?
- 100 (R1 G1/0) + 1 (R2 L0) = 101

To CHANGE the OSPF COST of an INTERFACE, you use the command :

💡 R1(config-if)# ip ospf cost

MANUAL COSTS take precedent over AUTOMATIC CALCULATED 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, etcetera…
- 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
THEN use the **"ip ospf cost"** command to change the COST of the 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
```

SUMMARY:
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-router)# ip ospf cost

3. Change the ***interface bandwidth***

💡 R1(config-router)# **bandwidth <***kilobits-per-second>*

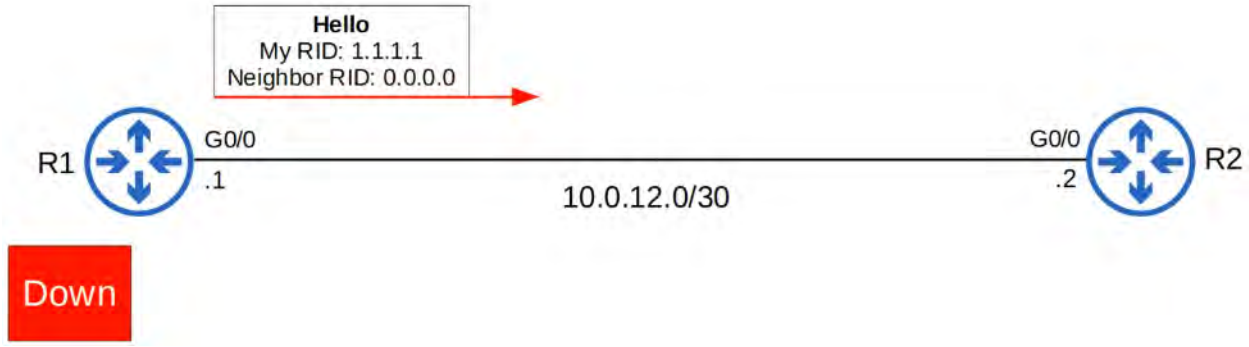
```
R3#show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Lo0          1     0               3.3.3.3/32         1     LOOP  0/0
Gi1/0        1     0               192.168.3.126/25   100   DR    0/0
Fa2/0        1     0               10.0.34.1/30       1000  BDR   1/1
Gi0/0        1     0               10.0.13.2/30       100   DR    1/1
```

BECOMING 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.

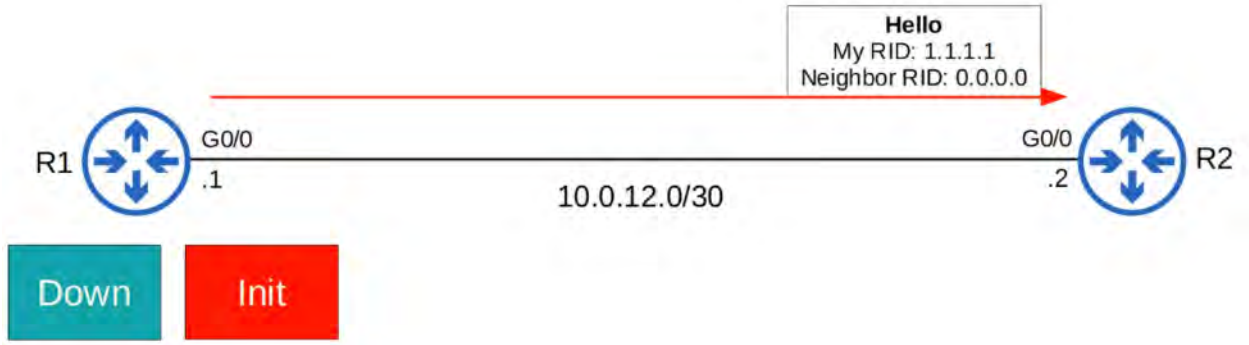
DOWN STATE

- OSPF is activated on R1s G0/0 INTERFACE
- It sends an OSPF "hello" message to 224.0.0.5
- It doesn't know about any OSPF neighbors yet, so the current neighbor state is DOWN



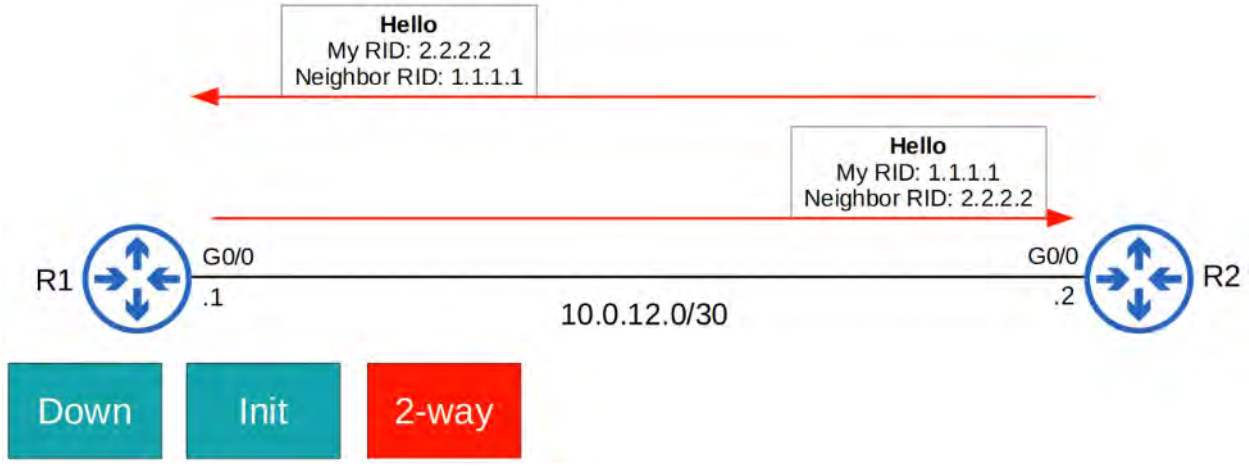
INIT STATE

- When R2 receives the “hello” packet, it will add an entry for R1 to its OSPF neighbor table
- In R2’s neighbor table, the relationship with R1 is now in the INIT state
- INIT state = “hello” packet received, but own ROUTER ID is not in the “hello” packet



2-WAY STATE

- R2 will send a “hello” packet containing the RID of BOTH ROUTERS
- R1 will insert R2 into its OSPF neighbor table in the 2-WAY state
- R1 will send another “hello” message, this time containing R2’s RID
- Both ROUTERS are now in the 2-WAY state



- The 2-WAY state means the ROUTER has received a “hello” packet with its own RID in it
- If both ROUTERS reach the 2-WAY state, it means that ALL of the conditions have been met for them to become OSPF neighbors.
- They are now READY to SHARE LSAs to build a common LSDB.
- In SOME NETWORK types, a DR (Designated ROUTER) and BDR (Backup Designated Router) will be elected at this point (OSPF Network Types and DR/DBR elections will be discussed in Day 28)

EXSTART STATE

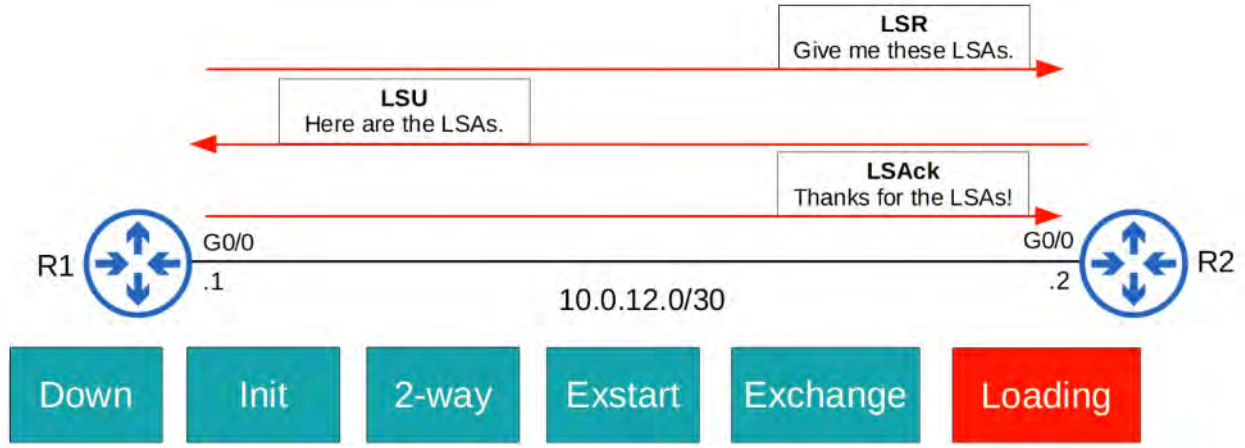
- The TWO ROUTERS will now prepare to exchange information about their LSDB
- Before that, they have to choose which one will START the exchange
- They do THIS in the EXSTART state
    - The ROUTER with the higher RID will become the MASTER and initiate the exchange.
    - The ROUTER with the lower RID will become the SLAVE
- To decide the MASTER and SLAVE, they exchange DBD (Database Description) packets

EXCHANGE STATE

- In the EXCHANGE state, the ROUTERS exchange DBDs which contain a LIST of the LSAs in their LSDB
- These DBDs do NOT include detailed information about the LSAs, just BASIC INFORMATION
- The ROUTERS compare the information in the DBD they received to the information in their OWN LSDB to determine which LSAs they must receive from their neighbor

LOADING STATE

- In the LOADING state, ROUTERS send **Link State Requests (LSR)** messages to request that their neighbors SEND them any LSAs they don't have
- LSAs are sent in **Link State Update (LSU)** messages
- The ROUTERS send **LSAck** messages to acknowledge that they received the LSAs

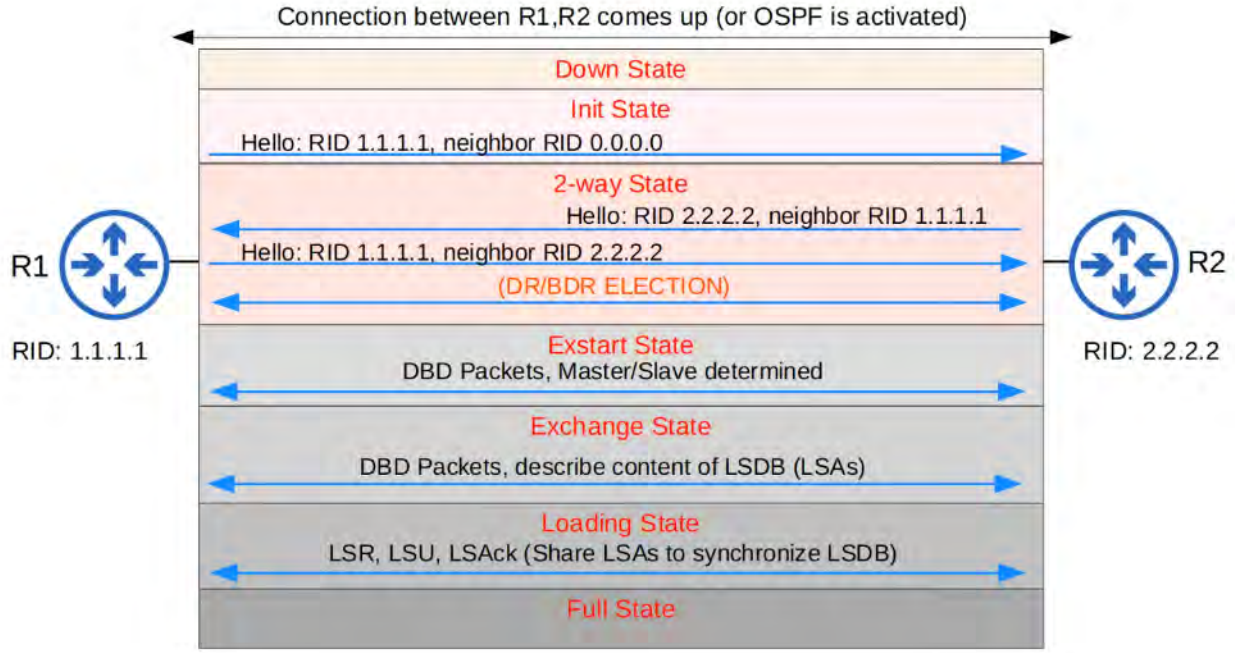


FULL STATE

- In the FULL state, the ROUTERS have a FULL OSPF adjacency and identical LSDBs

- They continue to SEND and LISTEN for “hello” packets (every 10 seconds by default) to maintain the neighbor adjacency
- Every time a “hello” packet is received, the “DEAD” timer (40 seconds by default) is reset
- If the DEAD timer counts down to 0 and no “hello” message is received, the neighbor is REMOVED
- The ROUTERS will continue to share LSAs as the network changes to make sure each ROUTER has a COMPLETE and ACCURATE map of the NETWORK (LSDB)

OSPF NEIGHBORS SUMMARY:



1) BECOME NEIGHBORS

- DOWN STATE
- INIT STATE
- 2-WAY STATE
- (DR/BDR ELECTION)

2. EXCHANGE LSAs

- EXSTART STATE
- EXCHANGE STATE
- LOADING STATE

SUMMARY OF OSPF MESSAGE TYPES

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.

MORE OSPF CONFIGURATIONS

Activate OSPF DIRECTLY on an INTERFACE with this command:

💡 R1(config-if)# ip ospf *process-id* area *area*

```
R1(config)#int g0/0
R1(config-if)#ip ospf 1 area 0
R1(config-if)#int g1/0
R1(config-if)#ip ospf 1 area 0
R1(config-if)#int g2/0
R1(config-if)#ip ospf 1 area 0
R1(config-if)#int l0
R1(config-if)#ip ospf 1 area 0
```

Configure ALL INTERFACES as OSPF Passive Interfaces

💡 R1(config-router) #passive-interface default

```
R1(config-if)#router ospf 1
R1(config-router)#passive-interface default
R1(config-router)#no passive-interface g0/0
R1(config-router)#no passive-interface g1/0
```

Can then REMOVE specific INTERFACES from being passive using:

💡 R1(config-router) #no passive-interface *interface-id*

Activating OSPF DIRECTLY on INTERFACES will show a different output in “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 1.1.1.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
  Routing on Interfaces Configured Explicitly (Area 0):
    Loopback0
    GigabitEthernet1/0
    GigabitEthernet0/0
    GigabitEthernet2/0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet2/0
    GigabitEthernet3/0
    Loopback0
    VoIP-Null0
  Routing Information Sources:
    Gateway         Distance      Last Update
    2.2.2.2              110      00:09:53
    Gateway         Distance      Last Update
    3.3.3.3              110      00:09:54
    4.4.4.4              110      00:09:54
  Distance: (default is 110)
```

They will appear under “Routing on Interfaces Configured Explicitly (Area #) :” (as above)

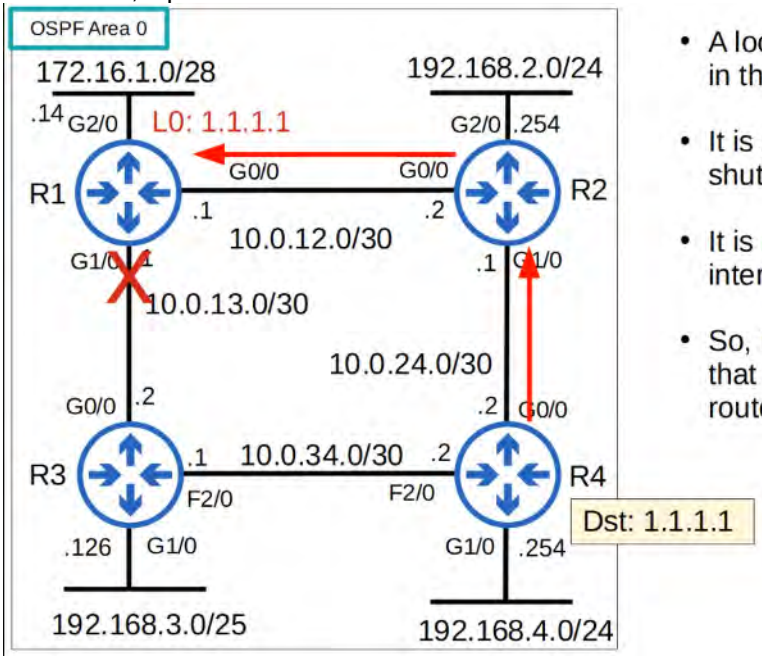
Showing the OSPF LSDB of a Device

```
R1(config-router)#do sh ip ospf database
            OSPF Router with ID (192.168.12.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
192.168.12.1    192.168.12.1    1119        0x80000003 0x002d45 3
192.168.34.1    192.168.34.1    1010        0x80000007 0x00ee10 2
192.168.245.2   192.168.245.2   319         0x80000007 0x00603d 2
192.168.245.1   192.168.245.1   319         0x80000006 0x004a83 3
203.0.113.1     203.0.113.1     235         0x80000004 0x003291 2

                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
192.168.34.2    192.168.245.2   1248        0x80000002 0x003bdc
192.168.245.3   203.0.113.1     319         0x80000002 0x00bcc9

                Type-5 AS External Link States
Link ID         ADV Router      Age         Seq#       Checksum Tag
0.0.0.0         203.0.113.1     205         0x80000001 0x00d2c1 1
```

28. OSPF : PART 3 (IGP: LINK STATE)

LOOPBACK INTERFACES

- A LOOPBACK INTERFACE is a virtual INTERFACE in the ROUTER
- It is ALWAYS UP/UP - unless you manually shut it down
- It is NOT dependent on a PHYSICAL INTERFACE
- So, it provides a consistent IP ADDRESS that can be used to REACH / IDENTIFY the ROUTER



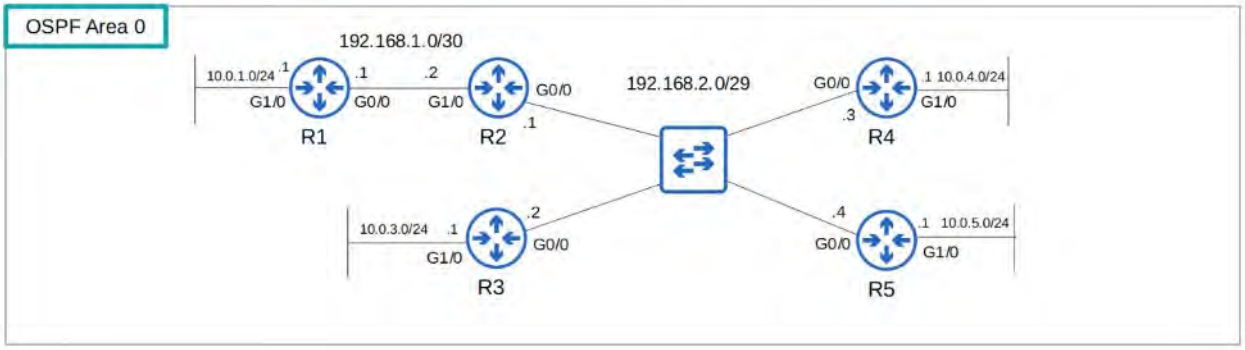
- A loopback interface is a virtual interface in the router.
- It is always up/up (unless you manually shut it down)
- It is not dependent on a physical interface.
- So, it provides a consistent IP address that can be used to reach/identify the router.

OSPF NETWORK TYPES

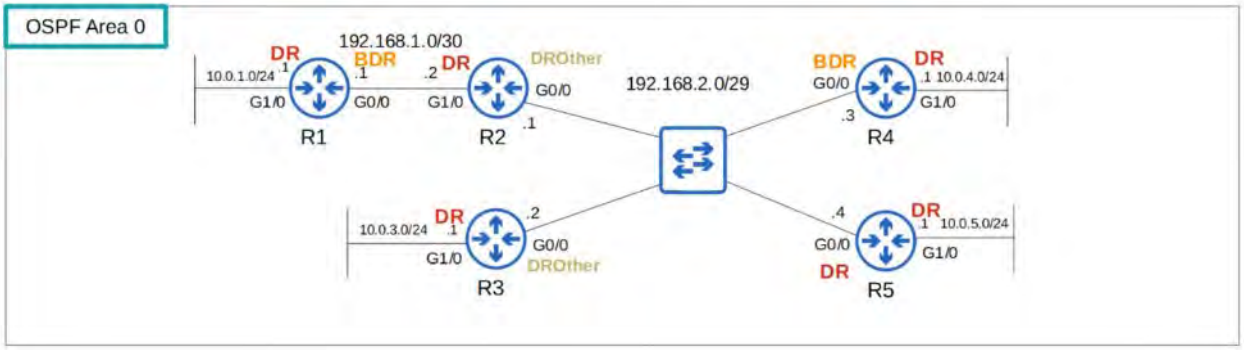
- The OSPF "NETWORK TYPE" refers to the TYPES of connection between OSPF neighbors (Ethernet, etc.)
- There are THREE MAIN OSPF NETWORK TYPES:
- BROADCAST :
  - Enabled by DEFAULT on **ETHERNET** and **FDDI** (Fiber Distributed Data Interfaces) INTERFACES
- POINT TO POINT :
  - Enabled by DEFAULT on **PPP** (Point-to-Point) and **HDLC** (High-Level Data Link Control) INTERFACES
- NON-BROADCAST :
  - Enabled by DEFAULT on **FRAME RELAY** and **X.25** INTERFACES

💡 CCNA focuses on BROADCAST and POINT-TO-POINT types

OSPF BROADCAST NETWORK TYPE



- Enabled on ETHERNET and FDDI interfaces by DEFAULT
- ROUTERS *dynamically discover* neighbors by SENDING / LISTENING for OSPF "Hello" messages using the multicast address 224.0.0.5
- A **DR (DESIGNATED ROUTER)** and **BDR (BACKUP DESIGNATION 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**



The DR / BDR election order of priority:

1. Highest OSPF INTERFACE PRIORITY
2. Highest OSPF ROUTER ID

"First Place" becomes the DR for the SUBNET
"Second Place" because the BDR

💡 DEFAULT OSPF INTERFACE PRIORITY is "1" on ALL INTERFACES!

The command to change the OSPF PRIORITY of an INTERFACE is :

💡 R2(config-if)# ip ospf priority

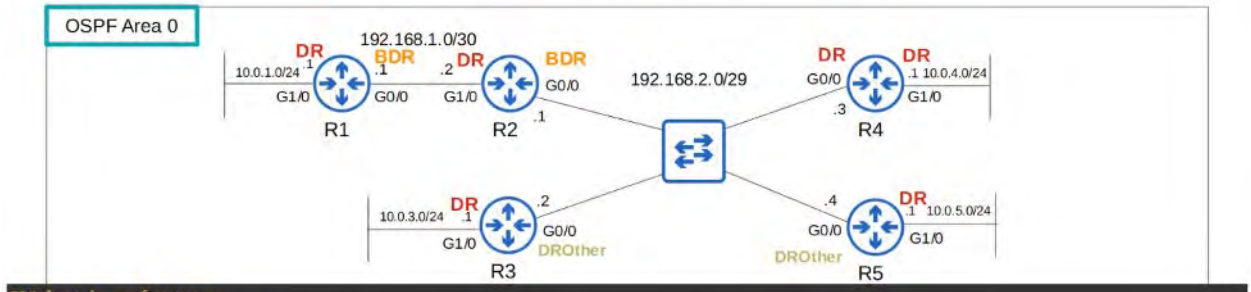
```
R2(config)#int g0/0
R2(config-if)#ip ospf priority ?
  <0-255>  Priority

R2(config-if)#ip ospf priority 255
```

💡 IF an OSPF PRIORITY is set to "0", the ROUTER CANNOT be the DR / BDR for the SUBNET!

The DR / DBR ELECTION is "non-preemptive".

Once the DR / DBR are selected, they will keep their role until OSPF is:

- Reset
- Interface fails
- Is shut down
- etc.




```
R5#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R5#
*Aug 22 04:25:05.307: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
*Aug 22 04:25:05.311: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
*Aug 22 04:25:05.311: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
R5#
*Aug 22 04:25:13.903: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
*Aug 22 04:25:13.907: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R5#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           255   FULL/BDR        00:00:37    192.168.2.1     GigabitEthernet0/0
3.3.3.3             1   2WAY/DROTHER    00:00:37    192.168.2.2     GigabitEthernet0/0
4.4.4.4             1   FULL/DR         00:00:39    192.168.2.3     GigabitEthernet0/0
```

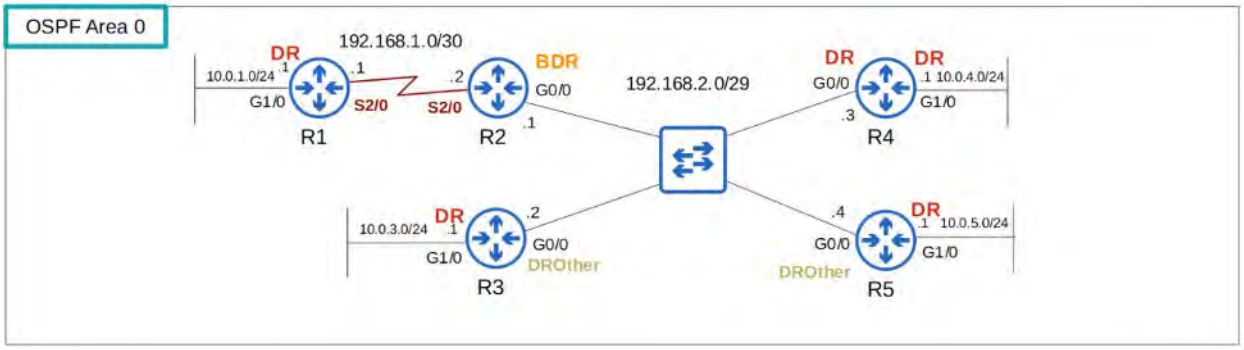
- **R4 became the DR, not R2. R2 became the BDR.**
  → When the DR goes down, the BDR becomes the new DR. Then an election is held for the next BDR.

- **R3 is a DROther, and is stable in the 2-way state.**
  → DROthers (R3 and R5 in this subnet) will only move to the FULL state with the DR and BDR. The neighbor state with other DROthers will be 2-way.

💡 In the BROADCAST NETWORK TYPE, ROUTERS will only form a FULL OSPF ADJACENCY with the DR and the 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
💡 MESSAGES to the DR / BDR are MULTICAST to 224.0.0.6
The DR and BDR will form a FULL ADJACENCY with ALL ROUTERS in the SUBNET
DROthers will form a FULL ADJACENCY ONLY with the DR / BDR !

```
R3#show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Gi0/0        1     0               192.168.2.2/29     1     DROTH 2/3
Gi1/0        1     0               10.0.3.1/24        1     DR    0/0
```

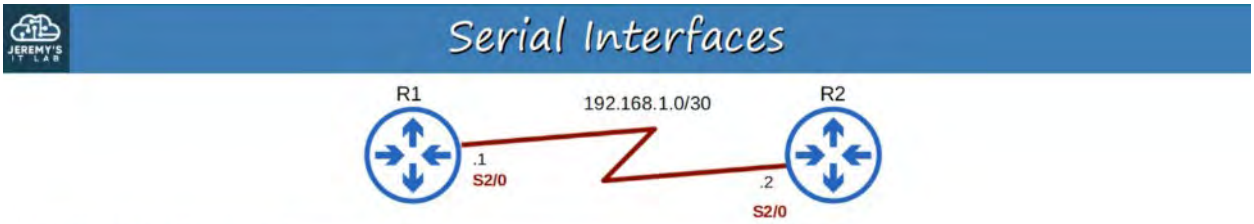
OSPF POINT-TO-POINT NETWORK TYPE



- ENABLED on **SERIAL** INTERFACES using the **PPP** and **HDLC** encapsulations, by DEFAULT
- ROUTERS dynamically discover neighbors by SENDING / LISTENING for OSPF “Hello” messages using the multicast address 224.0.0.5
- A DR and BDR are NOT elected
- These ENCAPSULATIONS are used for “Point-To-Point” connections
    - Therefore, there is no point in electing a DR and DBR
    - The TWO ROUTERS will form a FULL ADJACENCY with each other

---

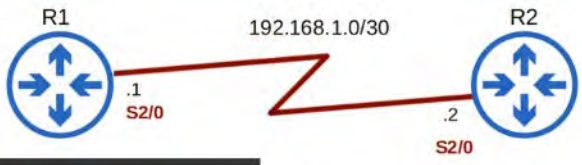
(ASIDE)
SERIAL INTERFACES




```
R1(config)#interface s2/0
R1(config-if)#clock rate ?
  With the exception of the following standard values not subject to rounding,

         1200 2400 4800 9600 14400 19200 28800 38400
         56000 64000 128000 2015232

  accepted clockrates will be bestfitted (rounded) to the nearest value
  supportable by the hardware.

  <246-8064000>    DCE clock rate (bits per second)

R1(config-if)#clock rate 64000
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shut
```

- One side of SERIAL CONNECTION functions as DCE (Data Communications Equipment)
- The OTHER side functions as DTE (Data Terminal Equipment)
- ONLY the DCE side needs to specify the *clock rate* (speed) of the connection

ETHERNET INTERFACES use the “speed” command to configure the operating speed.
SERIAL INTERFACES use the “clock rate” command




```
R1#show interface s2/0
Serial2/0 is up, line protocol is up
  Hardware is M4T
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC  crc 16, loopback not set
```

- The default encapsulation on a serial interface is HDLC.
  **actually cHDLC (Cisco HDLC)

cHDLC frame structure [edit]

The following table describes the structure of a cHDLC frame on the wire.[citation needed]

Address	Control	Protocol Code	Information	Frame Check Sequence (FCS)	Flag
8 bits	8 bits	16 bits	Variable length, 0 or more bits, in multiples of 8	16 bits	8 bits

- The Address field is used to specify the type of packet contained in the cHDLC frame; 0x0F for Unicast and 0x8F for Broadcast packets.
- The Control field is always set to zero (0x00).
- The Protocol Code field is used to specify the protocol type encapsulated within the cHDLC frame (e.g. 0x0800 for Internet Protocol).

If you change the ENCAPSULATION, it must MATCH on BOTH ENDS or the INTERFACE will go down.

```
R1(config)#int s2/0
R1(config-if)#encapsulation ppp
R1(config-if)#do show interface s2/0
Serial2/0 is up, line protocol is up
  Hardware is M4T
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP  LCP Open
```

R1 and R2 sharing the SAME Encapsulation Type

```
R1#show running-config interface s2/0
Building configuration...

Current configuration : 126 bytes
!
interface Serial2/0
 ip address 192.168.1.1 255.255.255.0
 encapsulation ppp
 serial restart-delay 0
 clock rate 64000
end
```

```
R2#show running-config interface s2/0
Building configuration...

Current configuration : 110 bytes
!
interface Serial2/0
 ip address 192.168.1.2 255.255.255.252
 encapsulation ppp
 serial restart-delay 0
end
```

SERIAL INTERFACES SUMMARY

- The DEFAULT encapsulation is HDLC
- You can configure PPP encapsulation with this command:

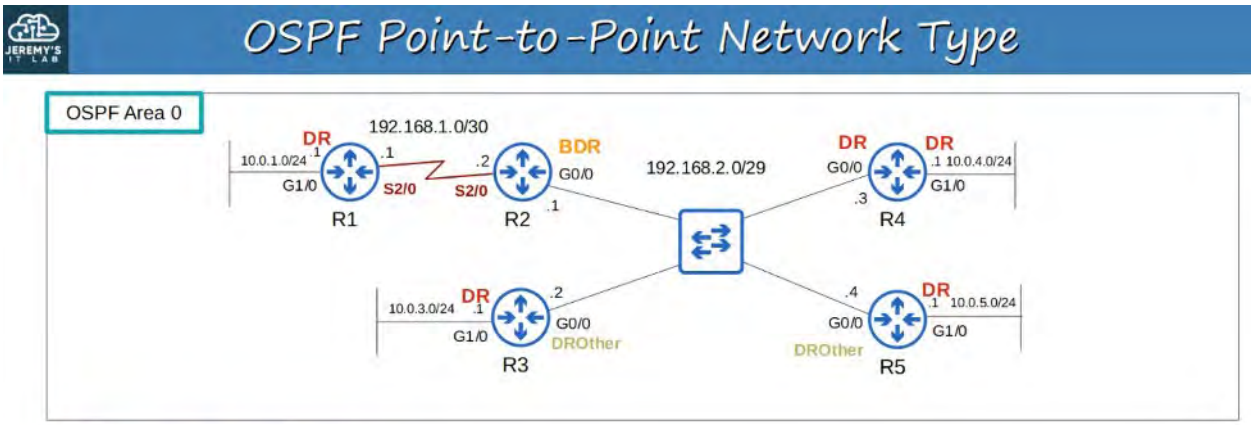
💡 R1(config-if)# **encapsulation ppp**

- One side is DCE, other 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*




```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           0   FULL/  -        00:00:31    192.168.1.1     Serial2/0
3.3.3.3           1   2WAY/DROTHER    00:00:39    192.168.2.2     GigabitEthernet0/0
4.4.4.4           1   FULL/DR         00:00:38    192.168.2.3     GigabitEthernet0/0
5.5.5.5           1   FULL/BDR        00:00:31    192.168.2.4     GigabitEthernet0/0
```

```
R1(config-if)#ip ospf network ?
  broadcast              Specify OSPF broadcast multi-access network
  non-broadcast          Specify OSPF NBMA network
  point-to-multipoint    Specify OSPF point-to-multipoint network
  point-to-point         Specify OSPF point-to-point network
```

- You can configure the OSPF NETWORK TYPE on an INTERFACE with :

💡 R1(config-if)# ip ospf network

For example, if TWO ROUTES are directly connected with an ETHERNET link, there is no need for a DR / DBR. You can configure the POINT-TO-POINT NETWORK type in this case

NOTE: Not all NETWORK TYPES work on ALL LINK TYPES (for example, a serial link cannot use the BROADCAST 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

💡 NON-BROADCAST NETWORK type Default Timers : Hello 30, Dead 120

OSPF NEIGHBOUR / ADJACENCY REQUIREMENTS

1. AREA NUMBER MUST MATCH

2. INTERFACES must be in the SAME SUBNET
3. OSPF PROCESS must not be **SHUTDOWN**

```
R2(config)#router ospf 1
R2(config-router)#shutdown
R2(config-router)#
*Aug 23 03:43:31.719: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Int
R2(config-router)#do show ip ospf neighbor
R2(config-router)#
```

4. OSPF ROUTER ID must be unique

```
R2(config-router)#router-id 192.168.1.1
% OSPF: Reload or use "clear ip ospf process" command, for this to take effect
R2(config-router)#end
R2#clear ip
*Aug 23 03:57:58.835: %SYS-5-CONFIG_I: Configured from console by console
R2#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R2#
*Aug 23 03:58:04.055: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or
R2#
*Aug 23 03:58:06.495: %OSPF-4-DUP_RTRID_NBR: OSPF detected duplicate router-id 192.168.1.1 from 192.168.1.1 on interface GigabitEthernet0/0
R2#show ip ospf neighbor
R2#
```

5. HELLO and DEAD Timers must MATCH
6. AUTHENTICATION settings must MATCH

```
R2(config-if)#ip ospf authentication-key jeremy
R2(config-if)#ip ospf authentication
R2(config-if)#
*Aug 23 04:56:28.435: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: D
R2(config-if)#do show ip ospf neighbor
R2(config-if)#
```

*** SPECIAL REQUIREMENTS ***

7. IP MTU settings must MATCH
- IP MTU : Maximum size of an IP Packet that can be sent from an INTERFACE
- If the settings DO NOT match, can still become OSPF Neighbors but OSPF WILL NOT operated properly

8. OSPF NETWORK TYPE must match
- will appear to be working but NEIGHBOR won't appear in ROUTING information

---

OSPF LSA TYPES

- The OSPF LSDB is made up of LSAs
- There are 11 types of LSA but there are only 3 you should be aware of for the CCNA:
    - Type 1 (Router LSA)
    - Type 2 (Network LSA)
    - Type 5 (AS External LSA)

TYPE 1 (Router LSA)

- Every OSPF ROUTER generates this type of LSA
- It identifies the ROUTER using it's ROUTER ID
- It also lists NETWORKS attached to the ROUTER's OSPF-Activated INTERFACES

TYPE 2 (Network LSA)

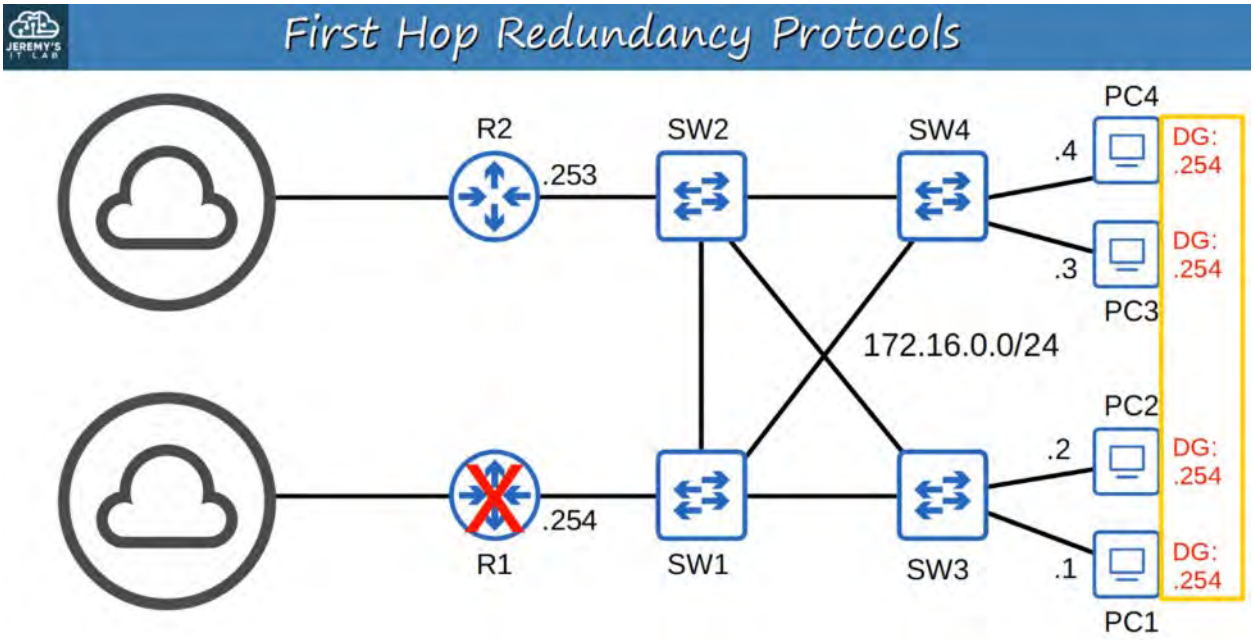
- Generated by the DR of EACH "multi-access" NETWORK (ie: the BROADCAST network type)
- Lists the ROUTERS which are attached to the multi-access NETWORK

TYPE 5 (AS-External LSA)

- Generated by ASBRs to describe ROUTES to DESTINATIONS outside of the AS (OSPF Domain)

29. FIRST HOP REDUNDANCY PROTOCOLS

THE PURPOSE OF FHRPS



What happens when the configured DEFAULT GATEWAY for network HOSTS goes down ?
What happens to the routed traffic?
How can we route our traffic to the functional GATEWAY at R2 (.253) ?
This is what the FIRST HOP REDUNDANCY PROTOCOL is designed to fix

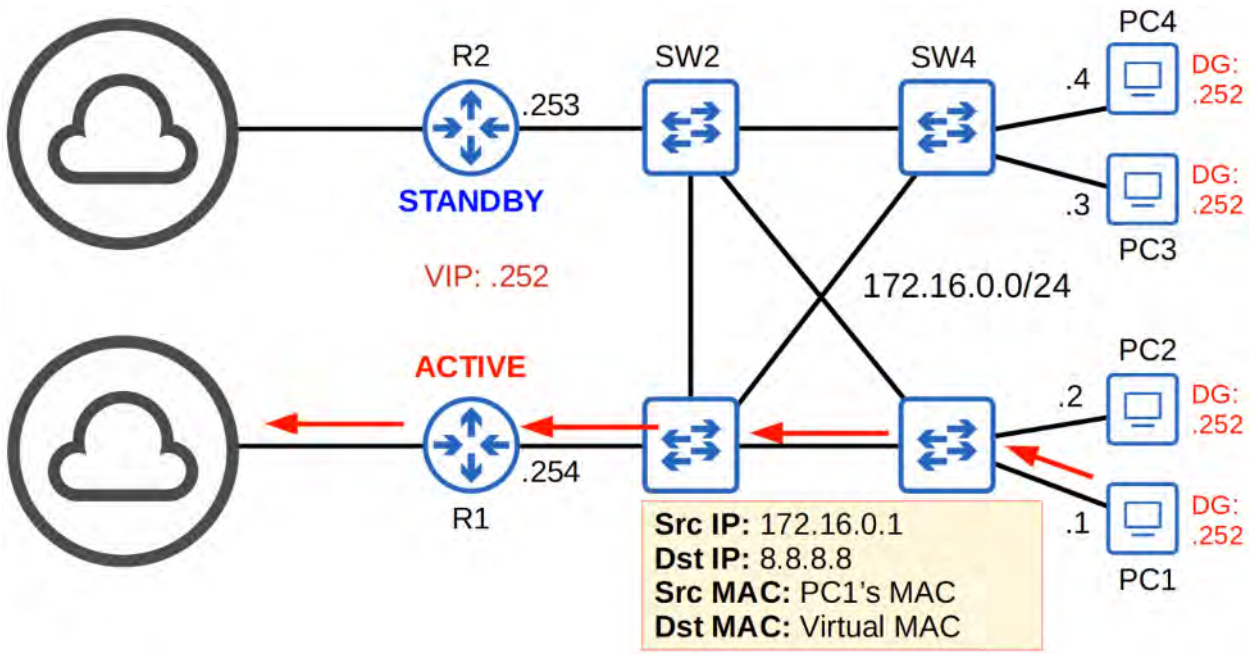
---

FIRST HOP REDUNDANCY PROTOCOL (FHRP)

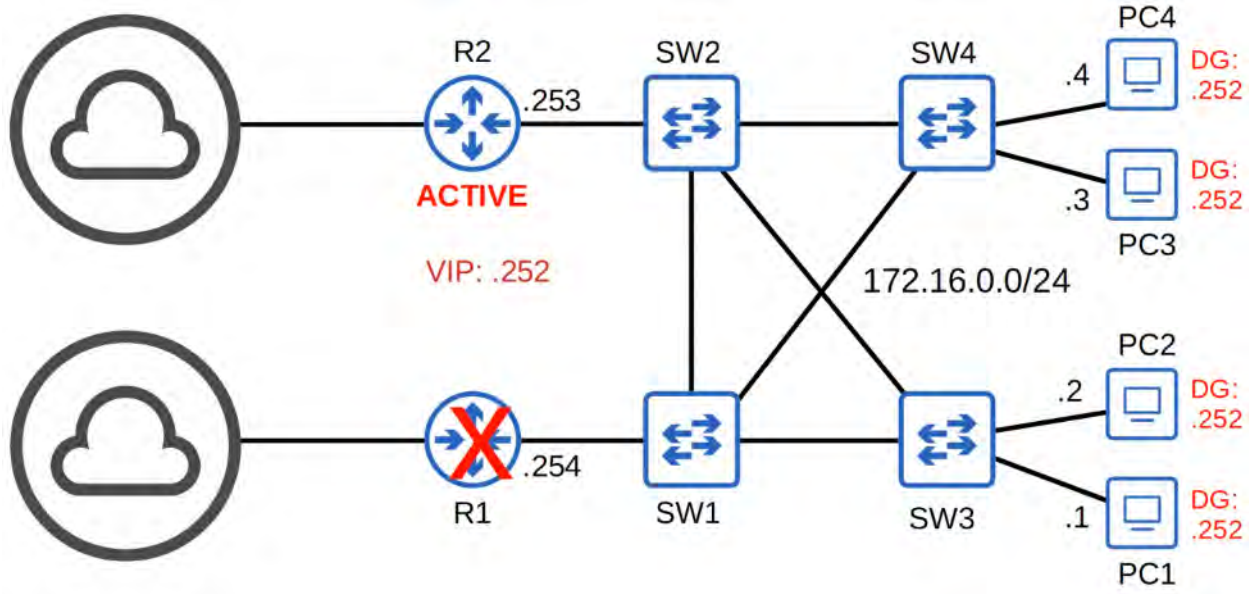
- Computer networking protocol
- Designed to PROTECT the DEFAULT GATEWAY used on a SUBNET by allowing TWO or MORE ROUTERS to provide BACKUP for that ADDRESS
- In the event of a FAILURE of the ACTIVE ROUTER, the BACKUP ROUTER will take over the ADDRESS (usually within seconds)

HOW DOES FHRP WORK?

- TWO (or more) ROUTERS share a VIP (A Virtual IP ADDRESS)
- THIS VIP is used by HOSTS as the DEFAULY GATEWAY IP
- The ROUTERS communicate with each other by sending “Hello” messages
- One ROUTER becomes the ACTIVE ROUTER, the other(s) STANDBY
- When a HOST sends traffic to an ADDRESS outside of the NETWORK, it sends an ARP REQUEST (Broadcast Flood) to the VIP to find out it’s MAC ADDRESS
    - Spanning Tree prevents BROADCAST STORM due to Broadcast Flood
- The ACTIVE ROUTER sends the ARP REPLY back (it’s VIRTUAL MAC ADDRESS) to the HOST
- The HOST now sends traffic OUTSIDE of the NETWORK with:
    - Source IP (HOST IP)
    - Destination IP (External IP ADDRESS)
    - Source MAC (HOST MAC ADDRESS)
    - Destination MAC (GATEWAY VIP MAC ADDRESS)



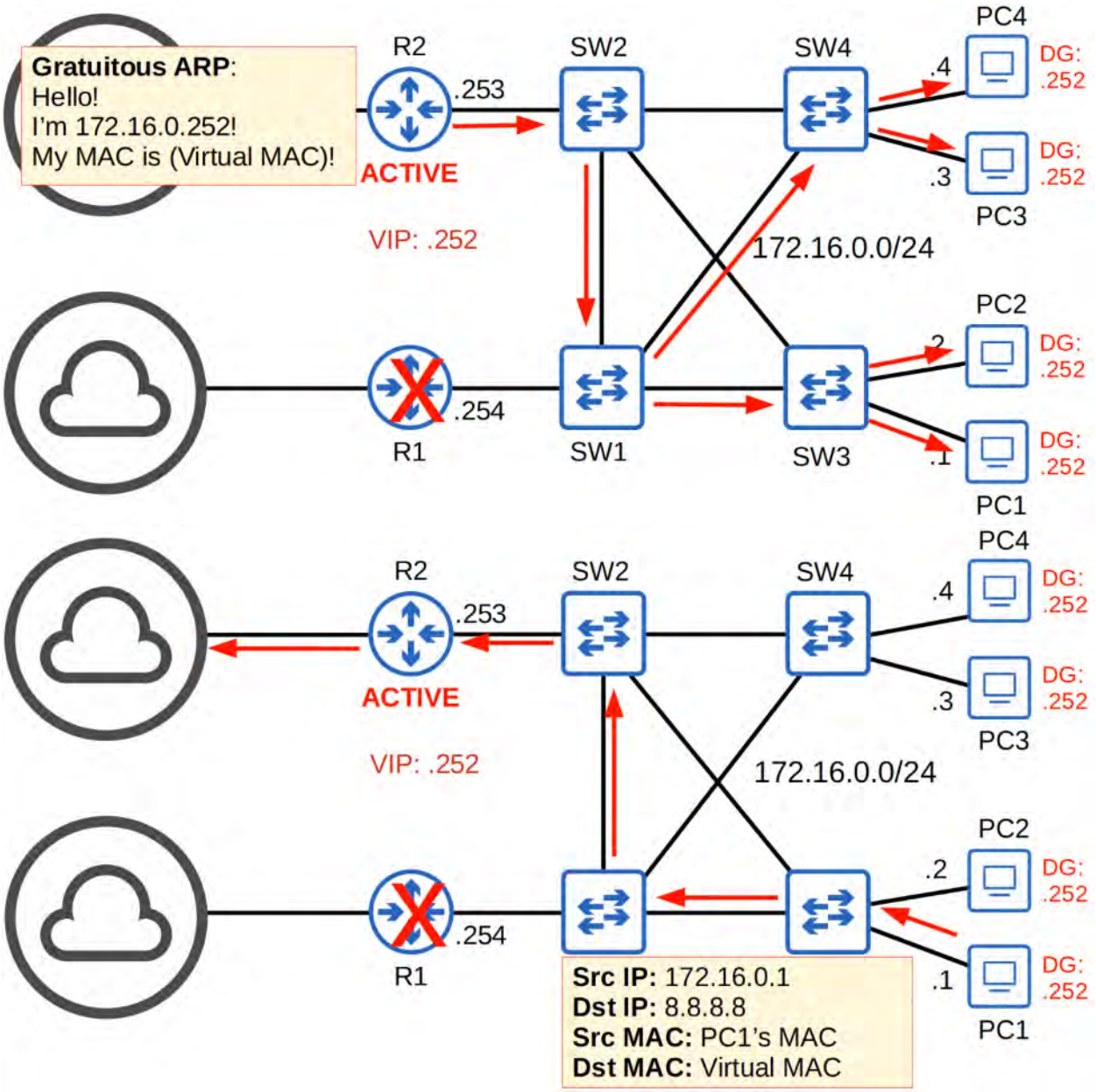
IF R1 goes down, R2 will switch from STANDY to ACTIVE after not receiving “Hello” messages from R1



The HOST ARP TABLE doesn’t need to change since the MAC ADDRESS of the VIP is already known and traffic flows externally via R2

R2 DOES need to update the SWITCHES with a GRATUITOUS ARP

- GRATUITOUS ARP is an ARP REPLY sent without being REQUESTED (no ARP REQUEST received)
- GRATUITOUS ARP uses BROADCAST (FFFF.FFFF.FFFF) - Normal ARP REPLY is Unicast



What happens is R1 comes back ONLINE again?
It becomes a STANDBY ROUTER
R2 remains the ACTIVE ROUTER

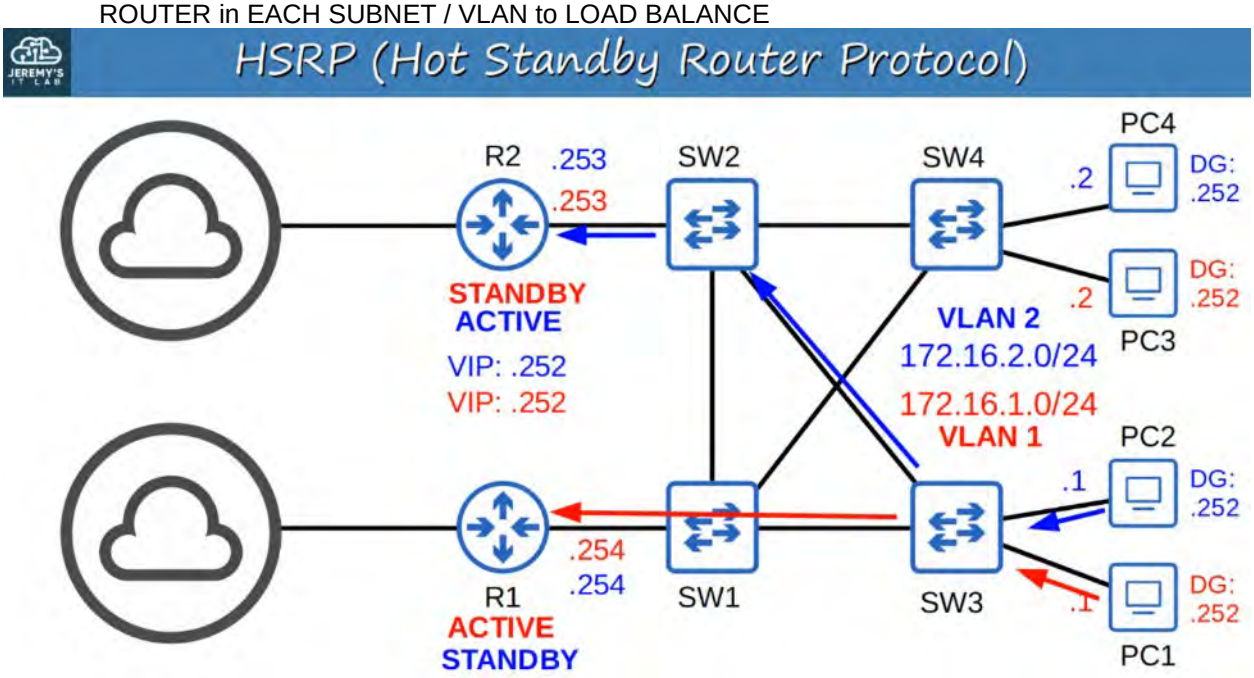
💡 FPRPs are "non-preemptive". The current ACTIVE ROUTER will not automatically give up its role, even if the former ACTIVE ROUTER returns.

*** You CAN change this setting to make R1 'preempt' R2 and take back it's ACTIVE role, automatically ***

HSRP (HOT STANDBY ROUTER PROTOCOL)

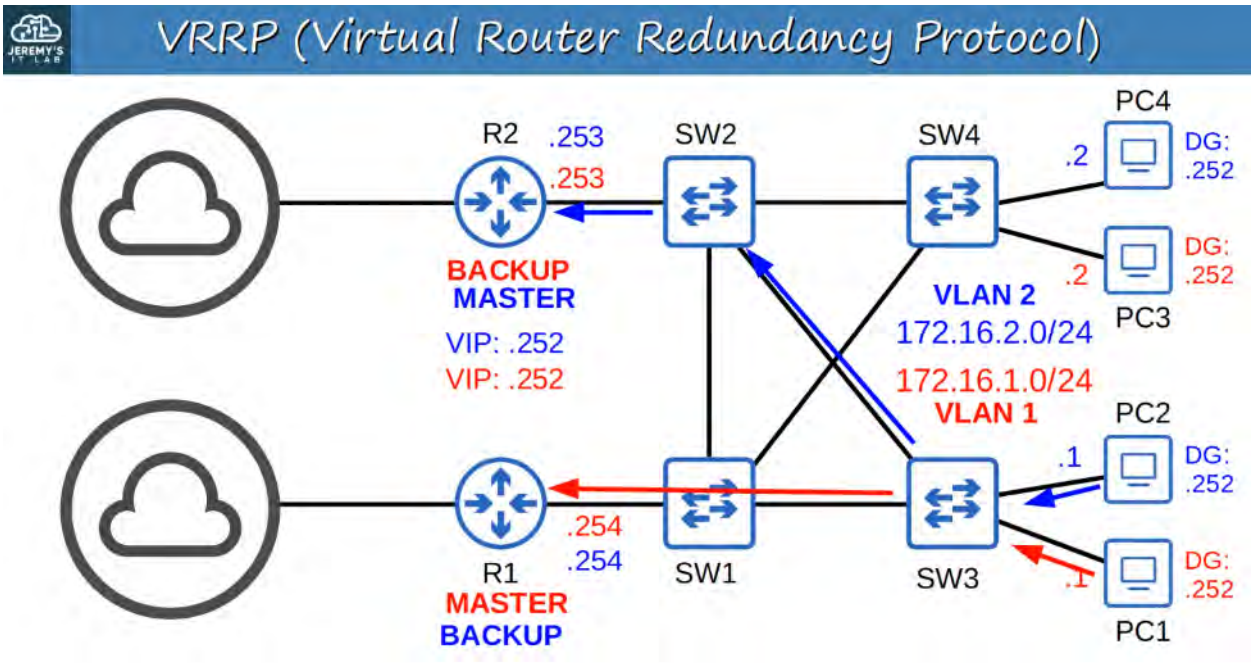
- **Cisco proprietary**
- An ACTIVE and STANDBY ROUTER are elected
- There are TWO VERSIONS :
  - version 1
  - version 2 : *adds IPv6 support* and increases # of *groups* that can be configured
- Multicast IPv4 ADDRESSES :

    - **v1** : 224.0.0.2
    - **v2** : 224.0.0.102
- VIRTUAL MAC ADDRESSES :
    - **v1** : 0000.0c07.acXX (XX = HSRP GROUP NUMBER)
    - **v2** : 0000.0c9f.fXXX (XXX = HSRP GROUP NUMBER)
- In a situation with MULTIPLE SUBNETS / VLANS, you can configure a DIFFERENT ACTIVE ROUTER in EACH SUBNET / VLAN to LOAD BALANCE



VRRP (VIRTUAL ROUTER REDUNDANCY PROTOCOL)

- Open Standard
- A MASTER and BACKUP ROUTER are elected
- Multicast IPv4 ADDRESSES :
    - 224.0.0.18
- VIRTUAL MAC ADDRESSES :
    - 0000.5e00.01XX (XX = VRRP GROUP NUMBER)
        - for GROUP NUMBERS > 99, you need to convert the number to HEX
        - Example: 200 = “c8” in Hex so the MAC would be 0000.5e00.01c8
- In a situation with MULTIPLE SUBNETS / VLANS, you can configure a DIFFERENT MASTER 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 can be an AVF, too)
- Each AVF acts as the DEFAULT GATEWAY for a portion of the HOSTS in the SUBNET
- Multicast IPv4 ADDRESSES :
    - 224.0.0.102
- VIRTUAL MAC ADDRESSES :
    - 0007.b400.XXYY (XX = GLBP GROUP NUMBER, YY = AVF NUMBER)

MEMORIZE THIS CHART and the differences between the FHRPs

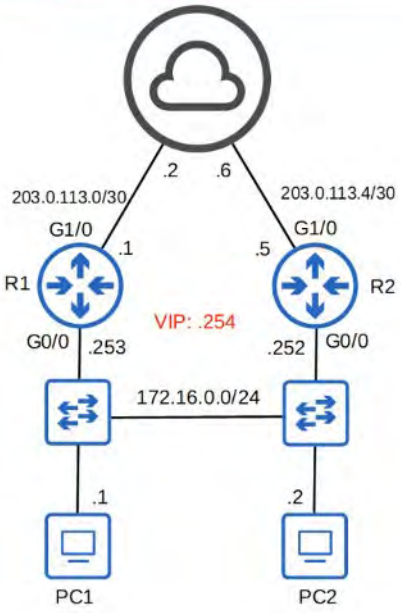
Comparing FHRPs

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

BASIC HSRP CONFIGURATION

R1s configuration

Configuring HSRP



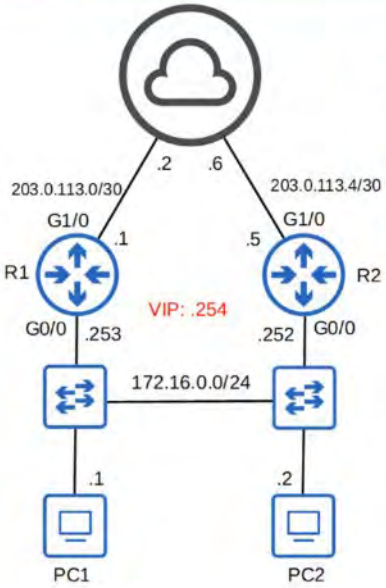

```
R1(config)#interface g0/0
R1(config-if)#standby ?
  <0-255>          group number
  authentication   Authentication
  bfd              Enable HSRP BFD
```

```
R1(config-if)#standby version 2
R1(config-if)#standby ?
  <0-4095>         group number
  authentication   Authentication
```

```
R1(config-if)#standby 1 ?
  authentication   Authentication
  follow           Name of HSRP group to follow
  ip               Enable HSRP IPv4 and set the virtual IP address
  ipv6             Enable HSRP IPv6
  mac-address      Virtual MAC address
  name             Redundancy name string
  preempt          Overthrow lower priority Active routers
  priority         Priority level
  timers           Hello and hold timers
  track            Priority tracking
```

NOTE : group number has to match ALL ROUTERS being configured in a given SUBNET

Configuring HSRP




```
R1(config-if)#standby 1 ip 172.16.0.254
R1(config-if)#
R1(config-if)#standby 1 priority ?
  <0-255>  Priority value

R1(config-if)#standby 1 priority 200
R1(config-if)#
R1(config-if)#standby 1 preempt
R1(config-if)#
```

The **active router** is determined in this order:
1 – Highest priority (default 100)
2 – Highest IP address

Preempt causes the router to take the role of active router, even if another router already has the role.

Only necessary on the router you want to become active

R2's configuration

```
R2(config-if)#standby version 2
R2(config-if)#
R2(config-if)#standby 1 ip 172.16.0.254
R2(config-if)#
R2(config-if)#standby 1 priority 50
R2(config-if)#
R2(config-if)#standby 1 preempt
```

HSRP version 1 and version 2 are not compatible.
If R1 uses version 2, R2 must use version 2 also.

NOTE : HSRP versions are not cross-compatible. All ROUTERS must use the same HSRP Version
Output of the “show standby” command

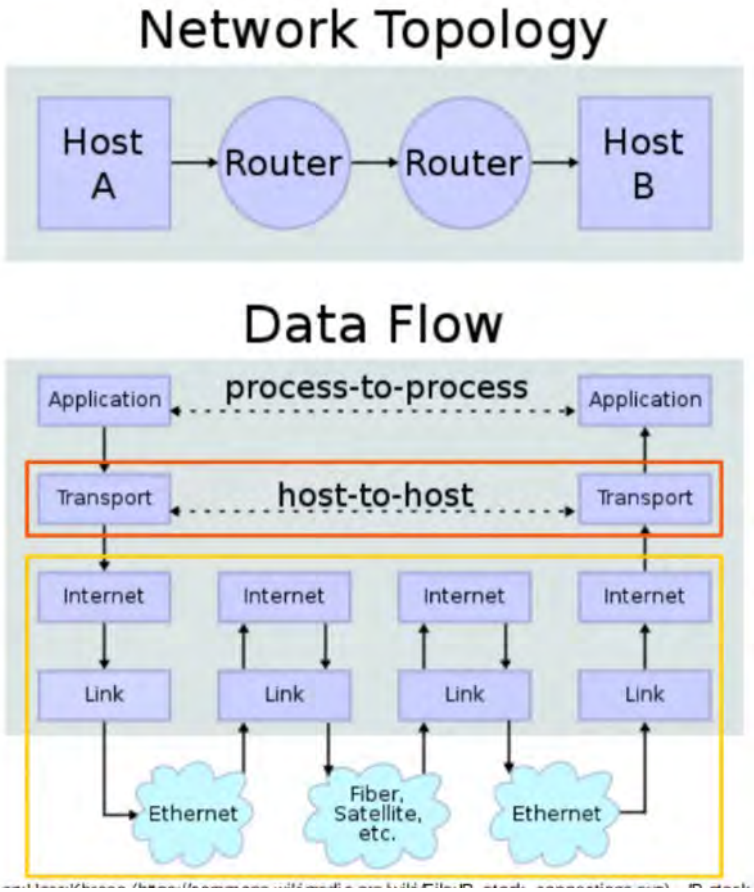
```
R1#show standby
GigabitEthernet0/0 - Group 1 (version 2)
  State is Active
    2 state changes, last state change 00:16:30
  Virtual IP address is 172.16.0.254
  Active virtual MAC address is 0000.0c9f.f001
    Local virtual MAC address is 0000.0c9f.f001 (v2 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.536 secs
  Preemption enabled
  Active router is local
  Standby router is 172.16.0.252, priority 50 (expires in 9.280 sec)
  Priority 200 (configured 200)
  Group name is "hsrp-Gi0/0-1" (default)
R1#
```

```
R2#show standby
GigabitEthernet0/0 - Group 1 (version 2)
  State is Standby
    1 state change, last state change 00:17:05
  Virtual IP address is 172.16.0.254
  Active virtual MAC address is 0000.0c9f.f001
    Local virtual MAC address is 0000.0c9f.f001 (v2 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.472 secs
  Preemption enabled
  Active router is 172.16.0.253, priority 200 (expires in 10.160 sec)
    MAC address is 0c9f.6041.8800
  Standby router is local
  Priority 50 (configured 50)
  Group name is "hsrp-Gi0/0-1" (default)
R2#
```

30. TCP and UDP (LAYER 4 PROTOCOLS)

BASICS OF LAYER 4

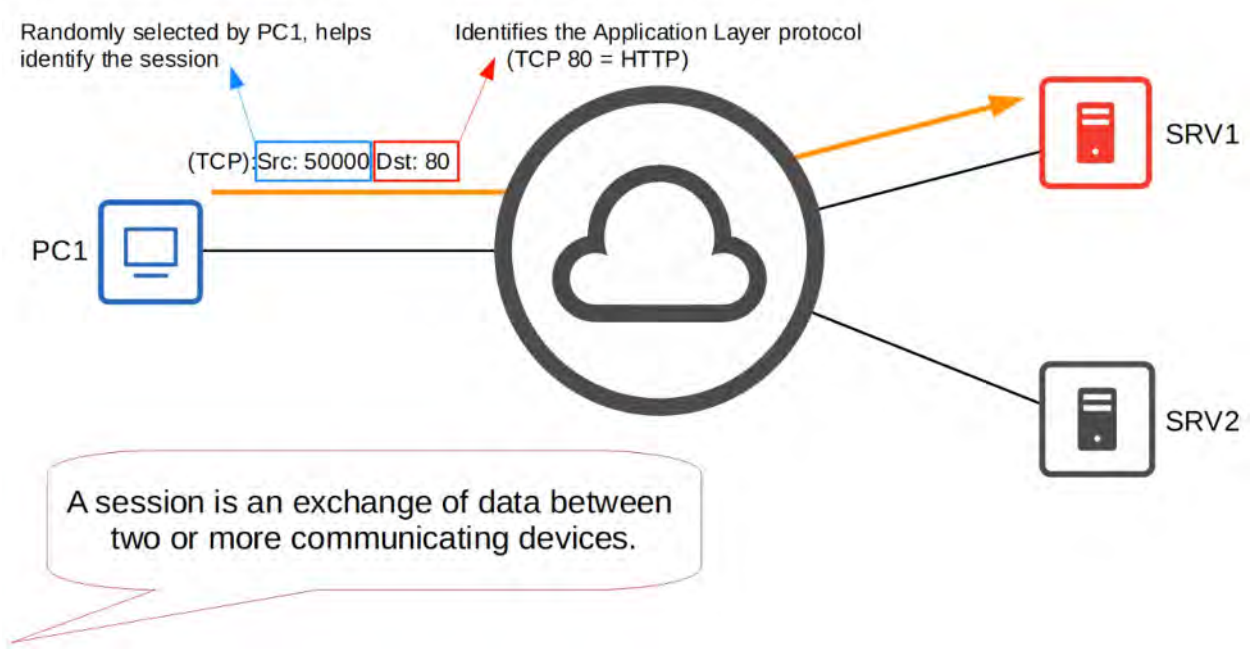
- Provides TRANSPARENT transfer of DATA between END HOSTS (Host To Host communication)



- Provides (or DOESN'T provide) various SERVICES to APPLICATIONS:
  - Reliable DATA Transfer
  - Error Recovery
  - Data Sequencing
  - Flow Control
- Provides LAYER 4 ADDRESSING (PORT numbers) - NOT the physical interfaces / ports on network devices
  - IDENTIFY the APPLICATION LAYER protocol
  - Provides SESSION multiplexing

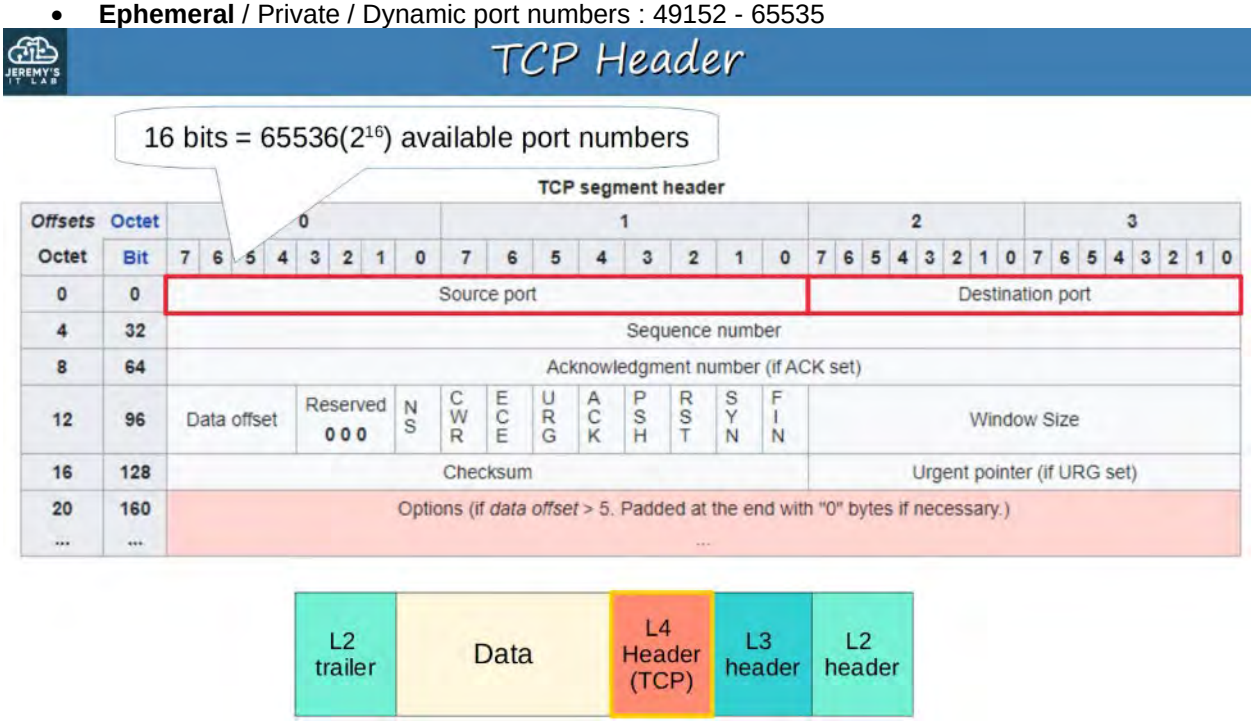
WHAT IS A SESSION ?

- A SESSION is an EXCHANGE of DATA between TWO or MORE communicating DEVICES



The FOLLOWING ranges have been designated by IANA (Internet Assigned Numbers Authority)

- **Well-Known** Port Numbers : 0 - 1023
- **Registered** Port Numbers : 1024 - 49151
- **Ephemeral** / Private / Dynamic port numbers : 49152 - 65535



TCP (TRANSMISSION CONTROL PROTOCOL)

- A CONNECTION-ORIENTED protocol
  - Before actually SENDING DATA to the DESTINATION HOST, the TWO HOSTS communicate to establish a CONNECTION. Once the CONNECTION is established, DATA exchange begins.

TCP segment header

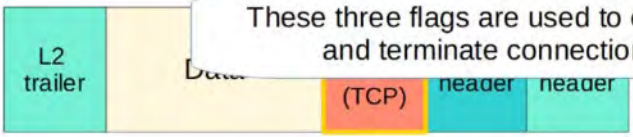
Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																	Urgent pointer (if URG set)														
20 ...	...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.)																															

These two fields provide sequencing and reliable communication.

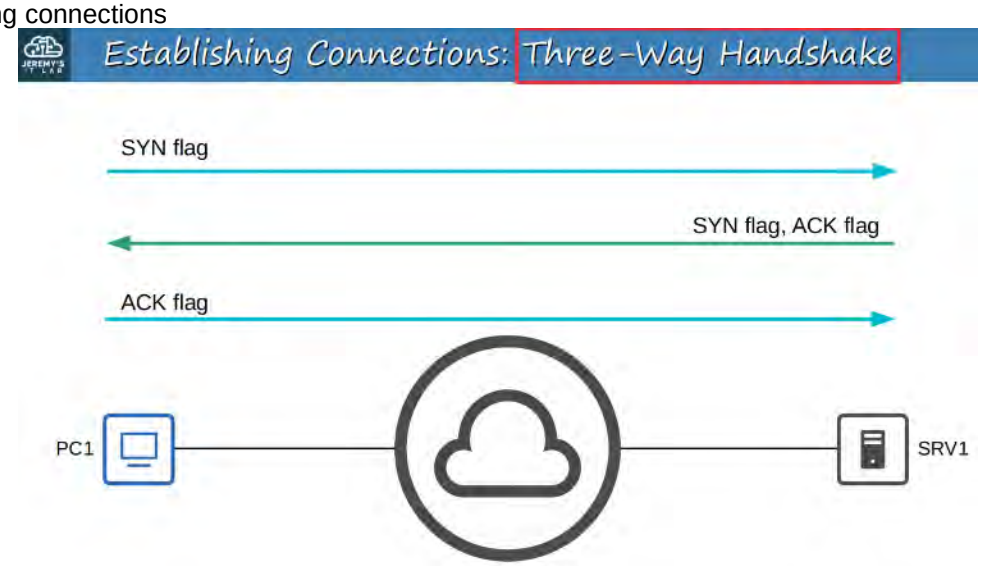
TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																	Urgent pointer (if URG set)														
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.)																															

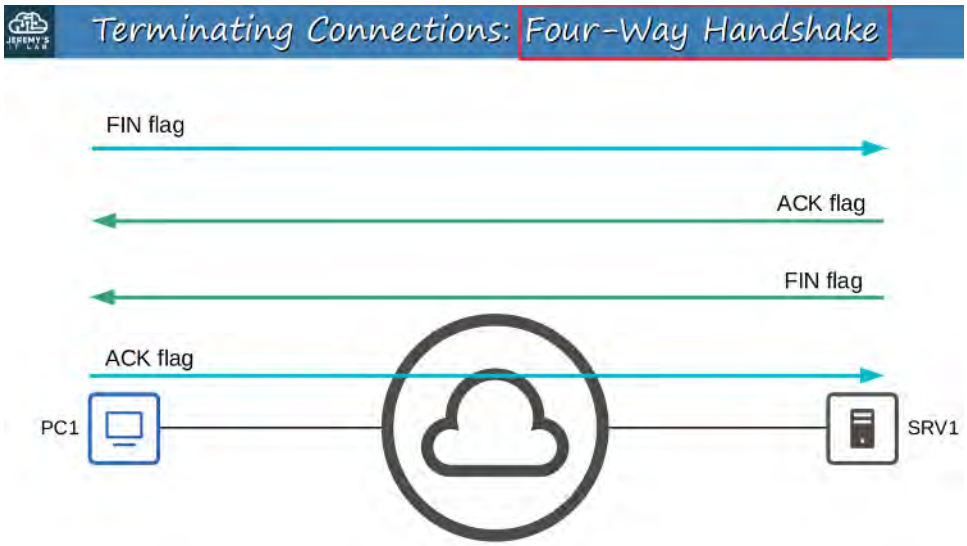
These three flags are used to establish and terminate connections.



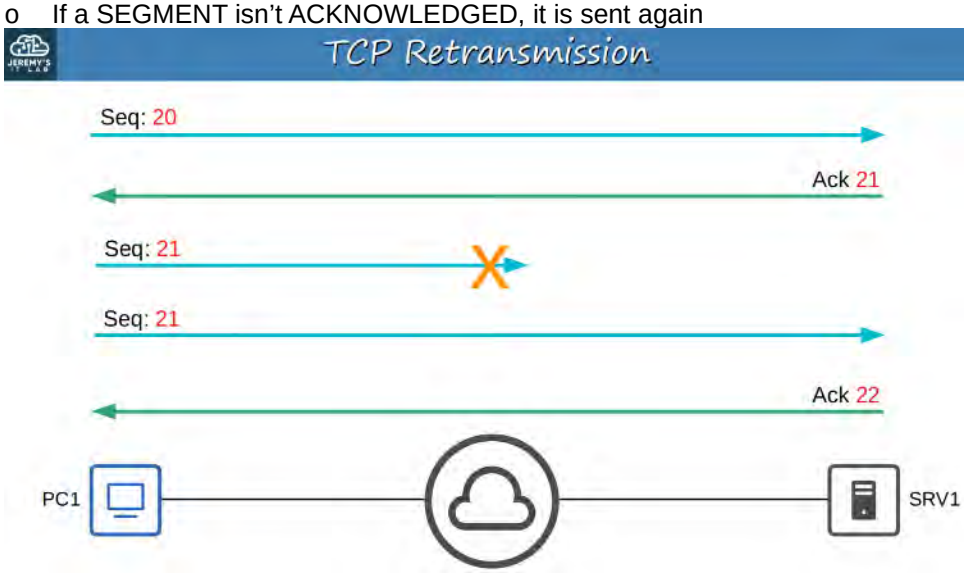
Establishing connections



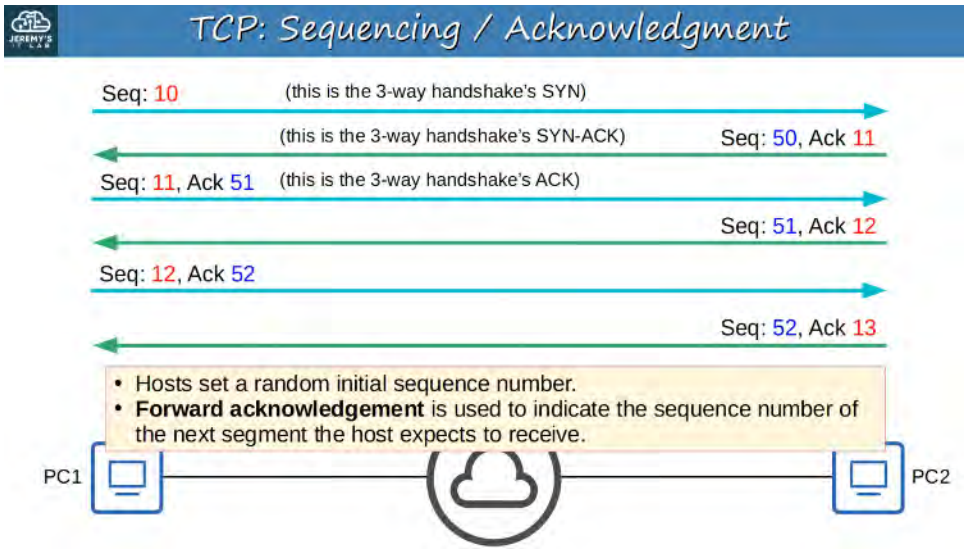
Terminating connections



- TCP provides RELIABLE communication
  - The DESTINATION HOST must acknowledge that it RECEIVED each TCP SEGMENT (Layer 4 PDU)
  - If a SEGMENT isn't ACKNOWLEDGED, it is sent again



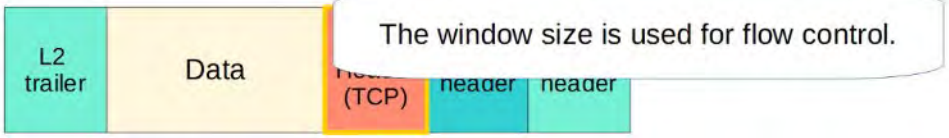
- TCP provides SEQUENCING
  - SEQUENCE numbers in the TCP HEADER allow DESTINATION HOSTS to put SEGMENTS in the correct ORDER even if they arrive out of ORDER



- TCP provides FLOW CONTROL
  - The DESTINATION HOST can tell the SOURCE HOST to increase / decrease the RATE that DATA is sent

TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																								Urgent pointer (if URG set)							
20	160	Options (if *data offset* > 5. Padded at the end w																ytes if necessary.)															
...	...	...																															



TCP Flow Control: Window Size

- Acknowledging every single segment, no matter what size, is inefficient.
- The TCP header's **Window Size** field allows more data to be sent before an acknowledgment is required.
- A 'sliding window' can be used to dynamically adjust how large the window size is.

In all of these examples, I used very simple sequence numbers. In real situations, the sequence numbers get much larger and do not increase by 1 with each message.
For the CCNA, just understand the concepts and don't worry about the exact numbers.

UDP (USER DATAGRAM PROTOCOL)

UDP datagram header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port														
4	32	Length																Checksum														

- UDP is NOT a CONNECTION-ORIENTED PROTOCOL
  - The SENDING HOST does NOT establish a CONNECTION with the DESTINATION HOST before sending DATA. The DATA is simply SENT
- UDP DOES NOT provide reliable COMMUNICATION
  - When UDP is used, ACKNOWLEDGEMENTS are NOT SENT for received SEGMENTS
  - If a SEGMENT is LOST, UDP has no mechanism to re-TRASMIT it
  - SEGMENTS are sent "best-effort"
- UDP DOES NOT provide SEQUENCING
  - There is NO SEQUENCE NUMBER FIELD in the UDP header
  - If SEGMENTS arrive out of order, UDP has no MECHANISM to put them back in ORDER
- UDP DOES NOT provide FLOW CONTROL
  - UDP has NO MECHANISM like TCP's WINDOW SIZE to control the flow of DATA
- UDP DOES provide ERROR CHECKING (via CHECKSUM)

COMPARING TCP AND UDP
Number of Fields in their Headers

Comparing TCP & UDP

TCP segment header

Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																								Urgent pointer (if URG set)							
20	160	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.)																															
...	...	...																															

UDP datagram header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

- TCP provides MORE FEATURES than UDP but at a COST of ADDITIONAL OVERHEAD
- For applications that require RELIABLE communications (for example, downloading a file), TCP is PREFERRED
- For applications, like real-time voice and video, UDP is preferred
- There are SOME applications that use UDP, but provide RELIABILITY, etc. within the APPLICATION itself.
- Some applications use BOTH TCP and UDP, depending on the situation.

TCP	UDP
Connection-oriented	Connectionless
Reliable	Unreliable
Sequencing	No sequencing
Flow control	No flow control
Use for downloads, file sharing, etc	Used for VoIP, live video, etc

IMPORTANT PORT NUMBERS

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)

31. IPv6 : PART 1

HEXIDECIMAL (Review)

Hexadecimal

- Binary / Base 2 / 0b
  0, 1

10 ← Is that decimal 10?
Or binary 10 (=decimal 2)?
Or hexadecimal 10 (=decimal 16)?

0b10

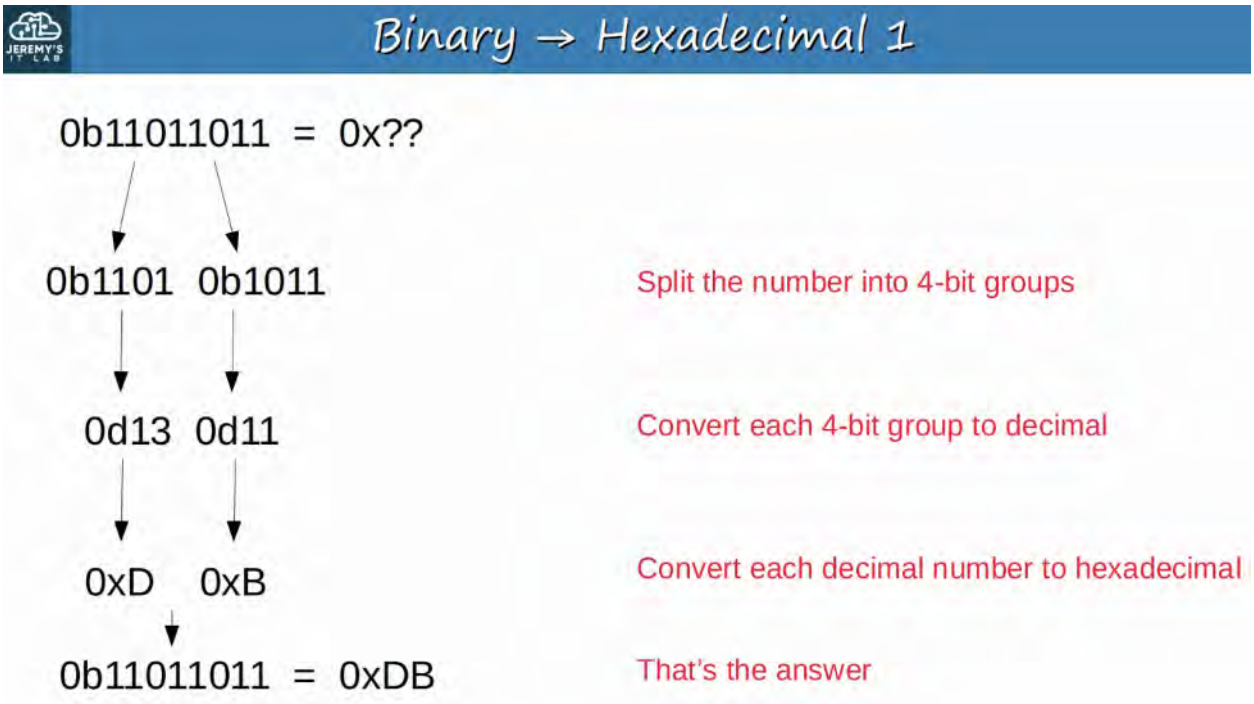
- Decimal / Base 10 / 0d
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9

- Hexadecimal / Base 16 / 0x
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

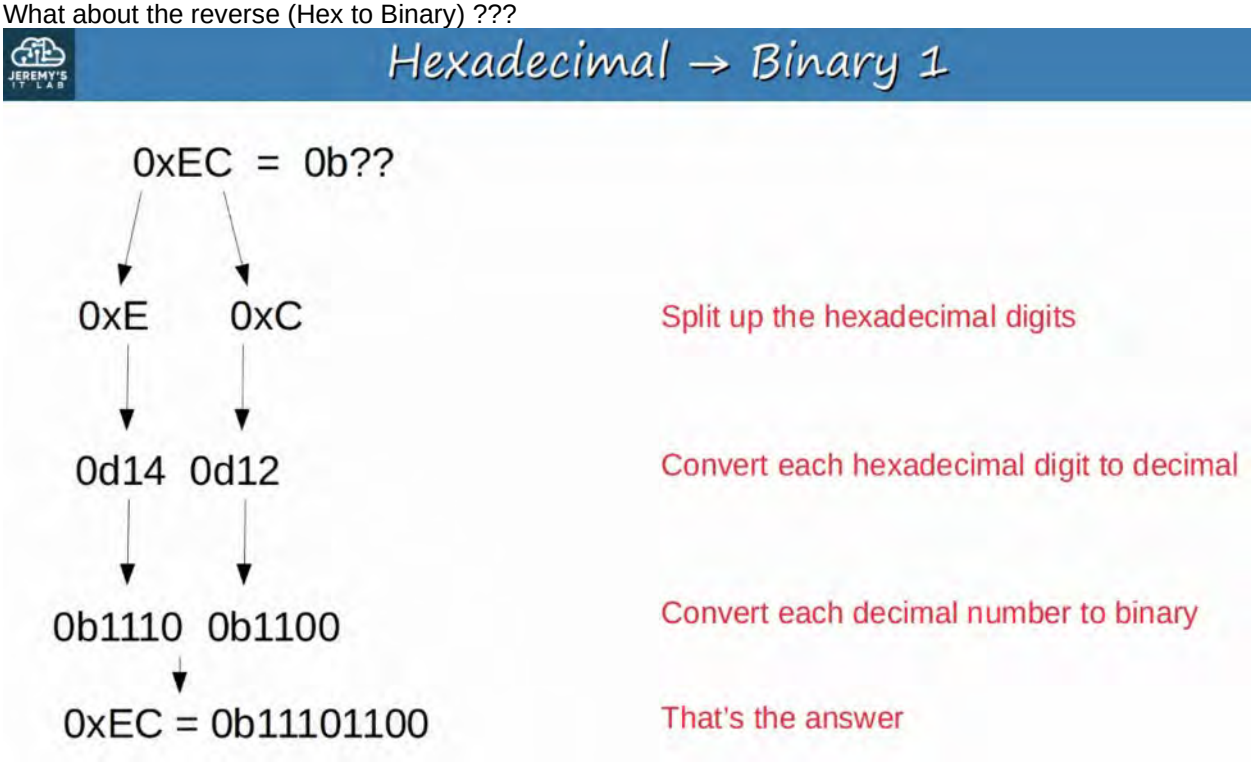
Hexadecimal

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9

Decimal	Binary	Hexadecimal
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



What about the reverse (Hex to Binary) ???



WHY IPv6?

- The **MAIN REASON** is that there are simply not enough IPv4 addresses available
- There are 2^32 IPv4 Addresses available (4,294,967,296)
- When IPv4 was being designed 30 years ago, the creators had NO idea the Internet would be as large as today
- VLSM, Private IPv4 ADDRESSES, and NAT have been used to conserve the use of IPv4 ADDRESS SPACE.

- These are short-term solutions, however.

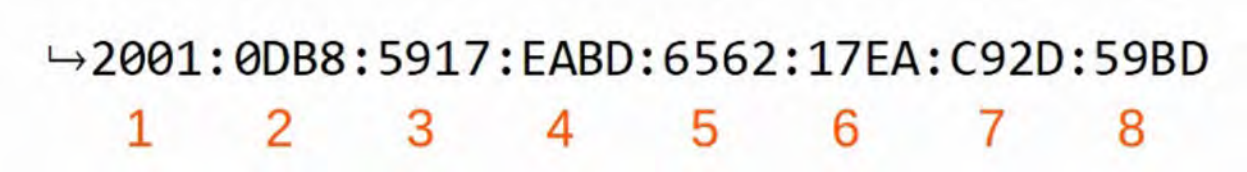
- The LONG -TERM solution is IPv6
- IPv4 ADDRESS assignments are controlled by IANA (Internet Assigned Number Authority)
- IANA distributes IPv4 ADDRESS space to various RIRs (Regional Internet Registries), which then assign them to companies that need them.

- On September 24th, 2015, ARIN declared exhaustion of the ARIN IPv4 address pool
- On August 21st, 2020, LACNIC announced that it had made its final IPv4 allocation

---

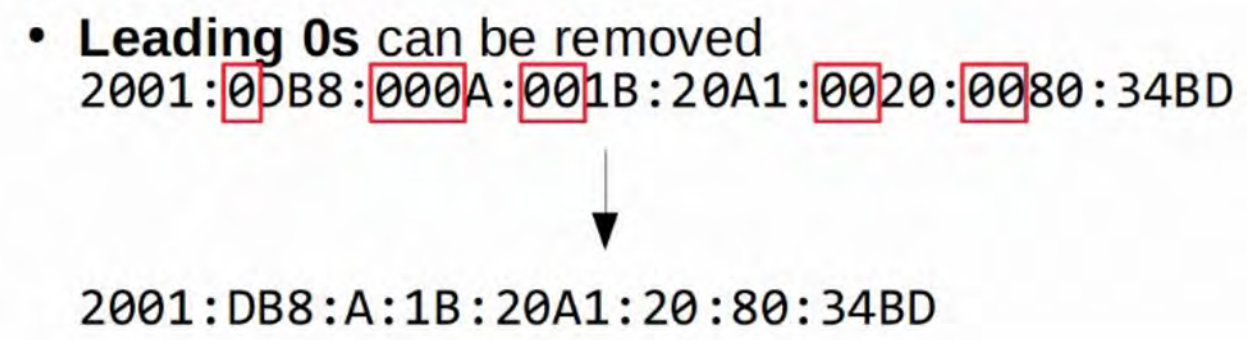
BASICS OF IPv6

- An IPv6 ADDRESS is **128 bits (8 bytes)**



- An IPv6 ADDRESS uses the / prefix number

SHORTENING (Abbreviating) IPv6 ADDRESSES



- **Consecutive quartets of all 0s** can be replaced with a double colon (::)

2001:0DB8:0000:0000:0000:0000:0080:34BD

↓

2001:0DB8::0080:34BD

↓ Combine both methods

2001:DB8::80:34BD

- Consecutive quartets of 0s can only be abbreviated once in an IPv6 address.

2001:0000:0000:0000:20A1:0000:0000:34BD

↓

~~2001::20A1::34BD~~

How many quartets of 0 are here?

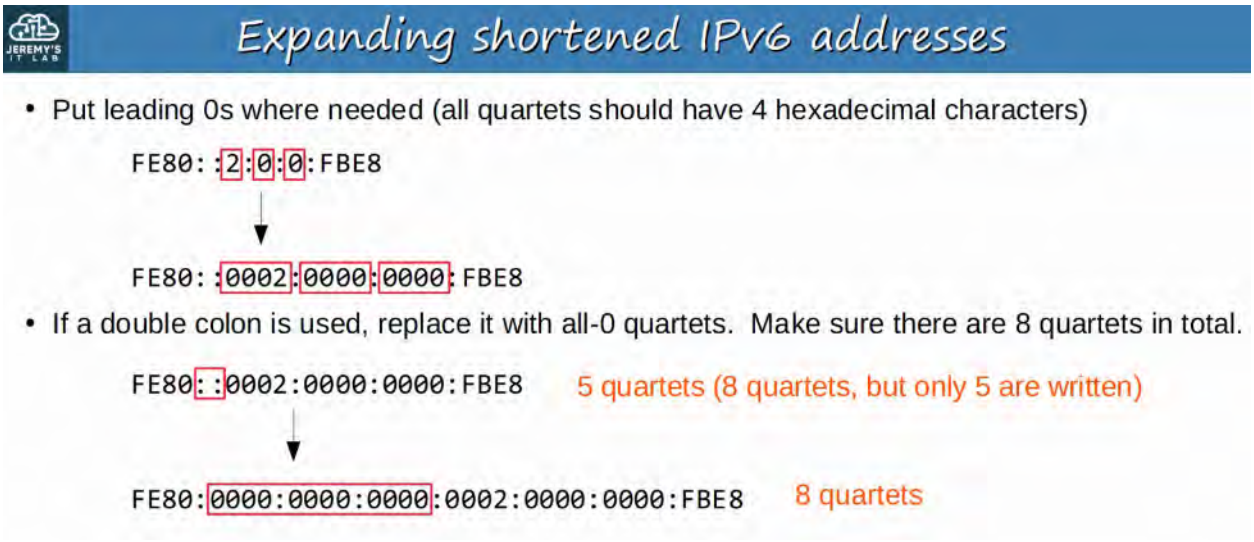
How many quartets of 0 are here?

↓

2001::20A1:0:0:34BD

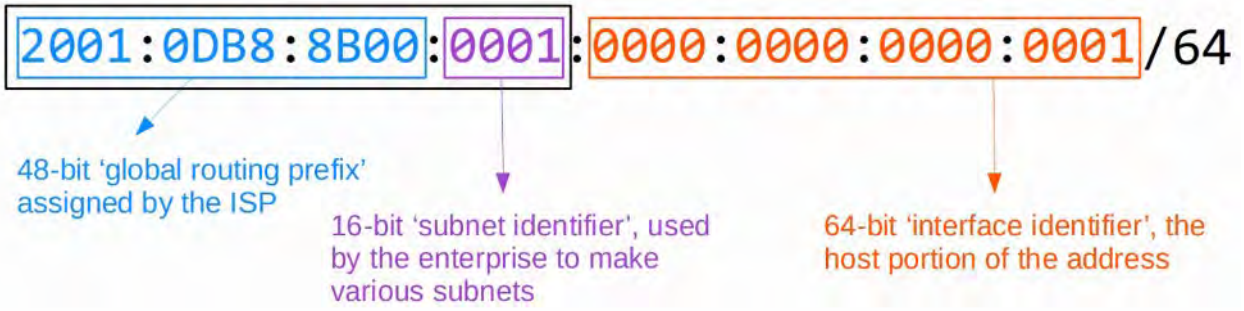
Full IPv6 Address	Shortened IPv6 Address
2000:AB78:0020:01BF:ED89:0000:0000:0001	2000:AB78:20:1BF:ED89::1
FE80:0000:0000:0000:0002:0000:0000:FBE8	FE80::2:0:0:FBE8
AE89:2100:01AC:00F0:0000:0000:0000:020F	AE89:2100:1AC:F0::20F
2001:0DB8:8B00:1000:0002:0BC0:0D07:0099	2001:DB8:8B00:1000:2:BC0:D07:99
2001:0DB8:0000:0000:0000:0000:0000:1000	2001:DB8::1000

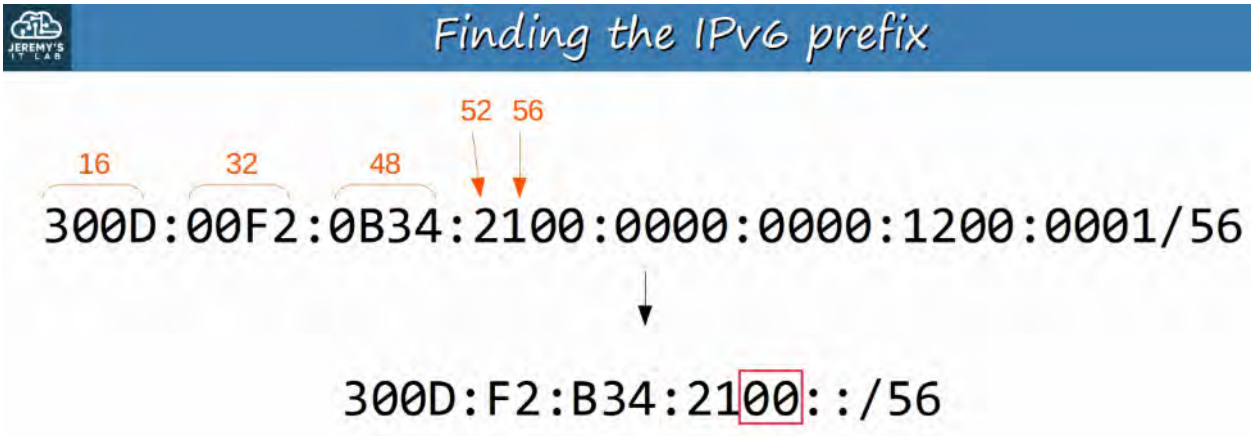
EXPANDING (Abbreviating) IPv6 ADDRESSES



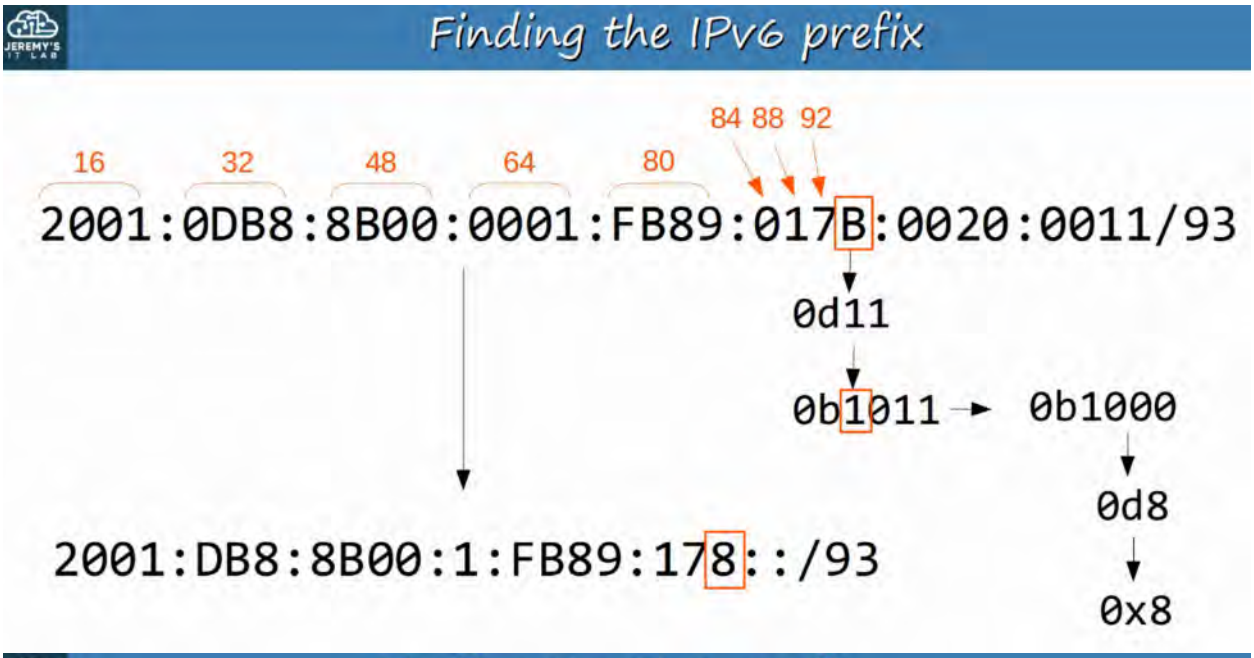
FINDING the IPv6 PREFIX (GLOBAL UNICAST ADDRESSES)

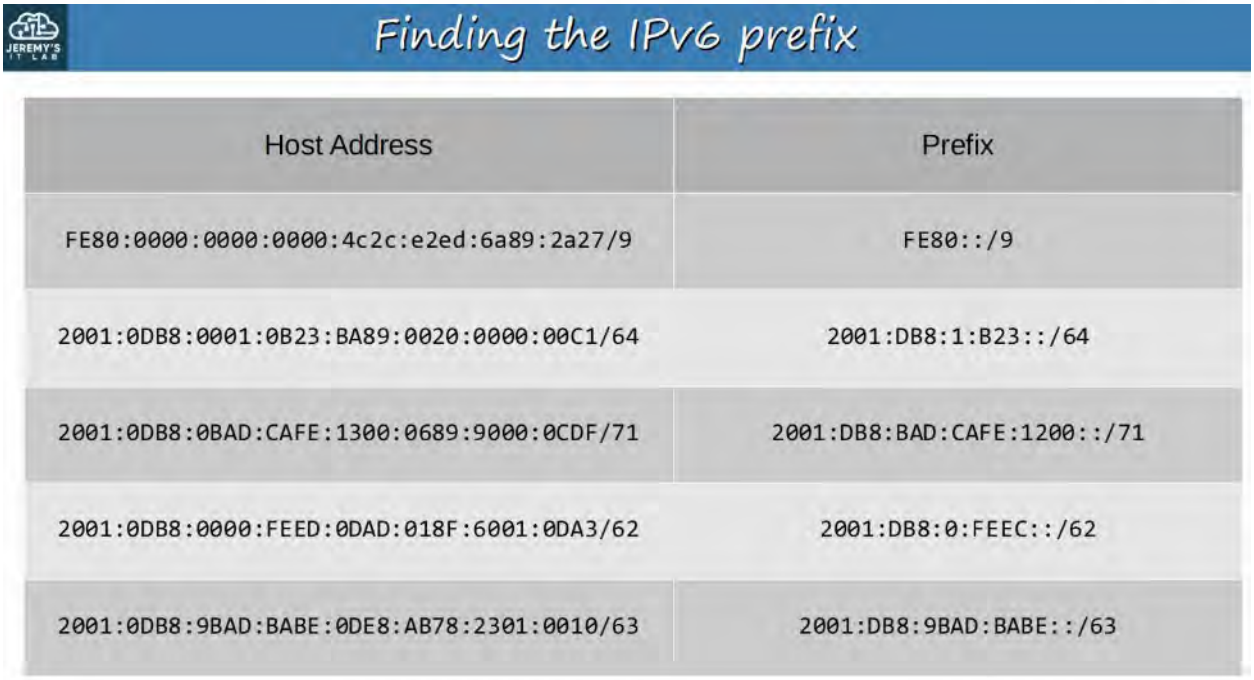
- Typically, an Enterprise requesting IPv6 ADDRESSES from their ISP will receive a /48 BLOCK
- Typically, IPv6 SUBNETS use a /64 PREFIX LENGTH
- That means an Enterprise has 16 bits to use to make SUBNETS
- The remaining 64 bits can be used for HOSTS



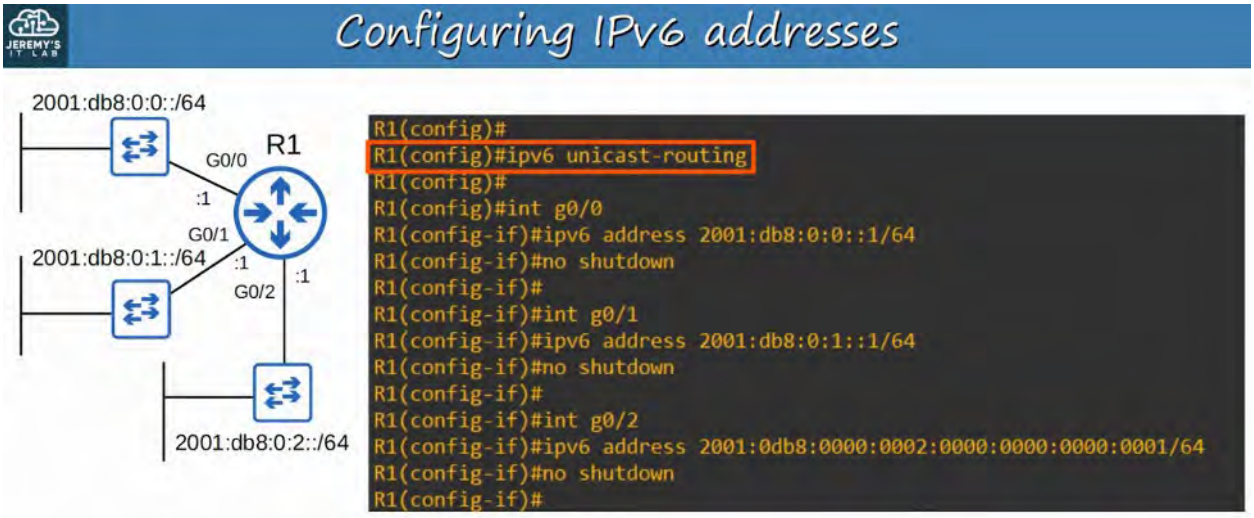


(Each digit is 4 bits / each 4 digit block is 16 bits)
REMEMBER : You can only remove the LEADING ZEROS !!!
2001 : 0DB8 : 8B00 : 0001 : FB89 : 017B : 0020 : 0011 /93
Because 93 lands in the middle of a 4 bit number, we need to convert the last digit to binary and borrow a "bit" from the first binary digit.
:: 017 [B] :: B = 0d11 = 0b1011 = 0b1000 (the first digit is borrowed, the remainder become 0)





Host Address	Prefix
FE80:0000:0000:0000:4c2c:e2ed:6a89:2a27/9	FE80::/9
2001:0DB8:0001:0B23:BA89:0020:0000:00C1/64	2001:DB8:1:B23::/64
2001:0DB8:0BAD:CAFE:1300:0689:9000:0CDF/71	2001:DB8:BAD:CAFE:1200::/71
2001:0DB8:0000:FEED:0DAD:018F:6001:0DA3/62	2001:DB8:0:FEEC::/62
2001:0DB8:9BAD:BABE:0DE8:AB78:2301:0010/63	2001:DB8:9BAD:BABE::/63



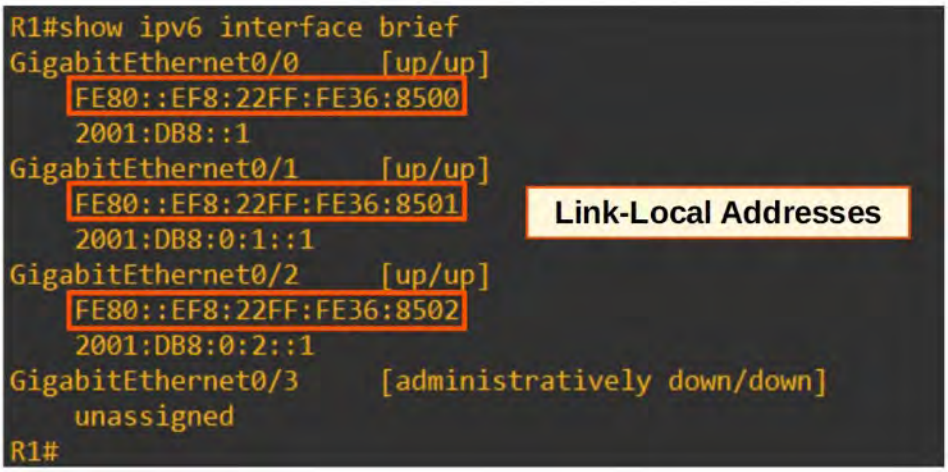
This allows the ROUTER to perform IPv6 ROUTING

💡 R1(config) #ipv6 unicast-routing

Configuring an INTERFACE with an IPv6 Address

💡 R1(config) #int g0/0 R1(config-if) #ipv6 address 2001:db8:0:0::1/64 R1(config) #no shutdown

You can also type out the full address (if necessary)



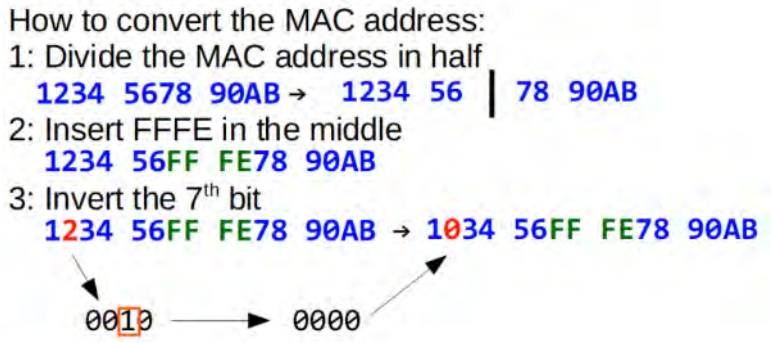
NOTE ABBREVIATED IPv6 ADDRESSES SHOWN

LINK-LOCAL ADDRESSES are automatically added when creating an IPv6 INTERFACE (Covered in IPv6 - PART 2 Lecture)

32. IPv6 : PART 2

IPv6 ADDRESS CONFIGURATION (EUI-64)

- EUI stands for Extended Unique Identifier
- (Modified) EUI-64 is a method of converting a MAC address (48-bits) into a 64-bit INTERFACE identifier
- This INTERFACE identifier can then become the “HOST portion” of a /64 IPv6 ADDRESS



EUI-64 PRACTICE:
782B CBAC 0867 >>> 782B CB || AC 0867

782B CBFF FEAC 0867

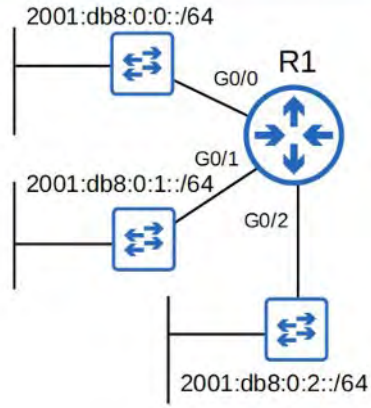
8 is the 7th bit so 1000 inverted becomes 1010 = A in hex

so the EUI-64 Interface Identifier is : 7A2B CBFF FEAC 0867

MAC Address	EUI-64 Interface Identifier
782B CBAC 0867	7A2B CBFF FEAC 0867
0200 4C4F 4F50	0000 4CFF FE4F 4F50
0050 56C0 0001	0250 56FF FEC0 0001
00FF 6BA6 F456	02FF 6BFF FEA6 F456
96AB 6D6B 98AE	94AB 6DFF FE6B 98AE

CONFIGURING IPv6 ADDRESSES with EUI-64

Configuring IPv6 addresses (EUI-64)




```
R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8::/64 eui-64
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#int g0/1
R1(config-if)#ipv6 address 2001:db8:0:1::/64 eui-64
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#int g0/2
R1(config-if)#ipv6 address 2001:db8:0:2::/64 eui-64
R1(config-if)#no shutdown
```

```
R1#show interfaces g0/0
GigabitEthernet0/0 is administratively down, line protocol is down
  Hardware is iGbE, address is 0cf8.2236.8500 (bia 0cf8.2236.8500)
```

```
R1#show interfaces g0/1
GigabitEthernet0/1 is administratively down, line protocol is down
  Hardware is iGbE, address is 0cf8.2236.8501 (bia 0cf8.2236.8501)
```

```
R1#show interfaces g0/2
GigabitEthernet0/2 is administratively down, line protocol is down
  Hardware is iGbE, address is 0cf8.2236.8502 (bia 0cf8.2236.8502)
```

```
R1(config-if)#do show ipv6 interface brief
GigabitEthernet0/0     [up/up]
    FE80::EF8:22FF:FE36:8500
    2001:DB8::EF8:22FF:FE36:8500
GigabitEthernet0/1     [up/up]
    FE80::EF8:22FF:FE36:8501
    2001:DB8:0:1:EF8:22FF:FE36:8501
GigabitEthernet0/2     [up/up]
    FE80::EF8:22FF:FE36:8502
    2001:DB8:0:2:EF8:22FF:FE36:8502
GigabitEthernet0/3     [administratively down/down]
    unassigned
```

NOTE the “2001:DB8…” Address has “E” changed to “c”. This is the 7th bit getting flipped (1110 to 1100 = 12 = hex ‘C’)

WHY INVERT THE 7th BIT ?

- MAC addresses can be divided into TWO TYPES:
  - UAA (Universally Administered Address)
    - Uniquely assigned to the device of the manufacturer
  - LAA (Locally Administered Address)
    - Manually assigned by an Admin (with the mac-address command on the INTERFACE) or protocol. Doesn’t have to be globally unique.

- You can INDENTIFY a UAA or LAA by the 7th bit of the MAC ADDRESS, called the U/L bit (Universal/Local bit)
    - U/L bit set to 0 = UAA
    - U/L bit set to 1 = LAA
- In the context of IPv6 addresses/EUI-64, the meaning of the U/L bit is reversed:
    - U/L bit set to 0 = The MAC address the EUI-64 INTERFACE ID was made from was an LAA
    - U/L bit set to 1 = The MAC address the EUI-64 INTERFACE ID was made from was a UAA

IPv6 ADDRESS TYPES

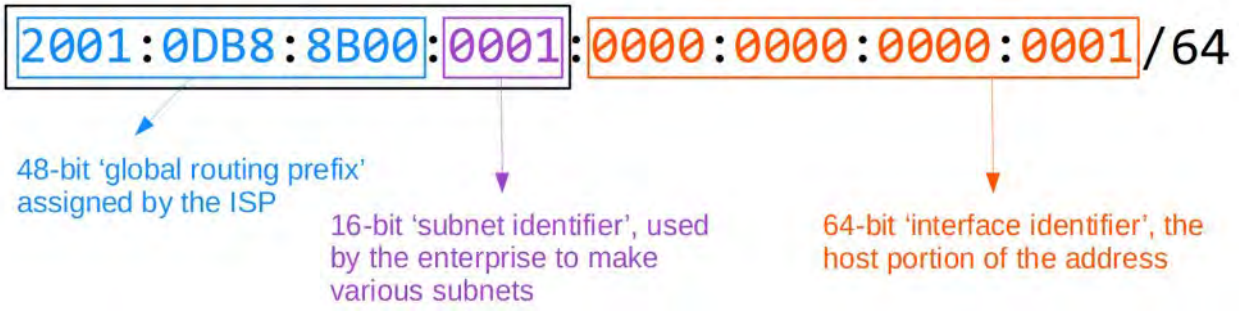
1. GLOBAL UNICAST ADDRESSES

- **Global Unicast** IPv6 ADDRESSES are PUBLIC ADDRESSES which can be used over the INTERNET
- Must REGISTER to use them.
- They are PUBLIC ADDRESSES so need to be GLOBALLY UNIQUE

💡 Originally defined as the 2000 :: /3 block (2000:: to 3FFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF)

- NOW defined as ALL ADDRESSES which are not RESERVED for other purposes

Remember THESE THREE PARTS of a GLOBAL UNICAST ADDRESS



---

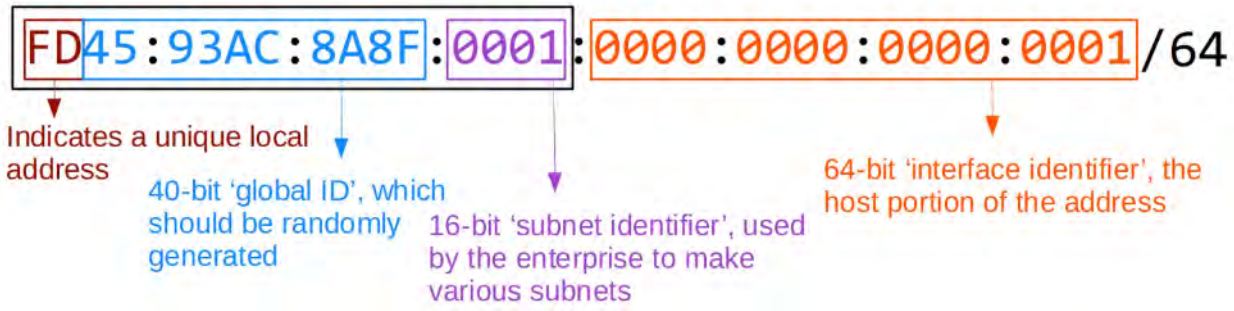
2. UNIQUE LOCAL ADDRESSES

- **Unique Local** IPv6 ADDRESSES are PRIVATE ADDRESSES which cannot be used over the internet
- You do NOT need to REGISTER to use them
- Can be used FREELY within INTERNAL NETWORKS
- Do NOT need to be GLOBALLY UNIQUE (*)
- CANNOT be ROUTED over the INTERNET

💡 Uses the ADDRESS block FC00 ::/7 (FC00:: to FDFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF)

- A later UPDATE required the 8th bit to be set to 1 so the FIRST TWO DIGITS must be FD

(*) The GLOBAL ID should be UNIQUE so that ADDRESSES don't overlap when companies MERGE



---

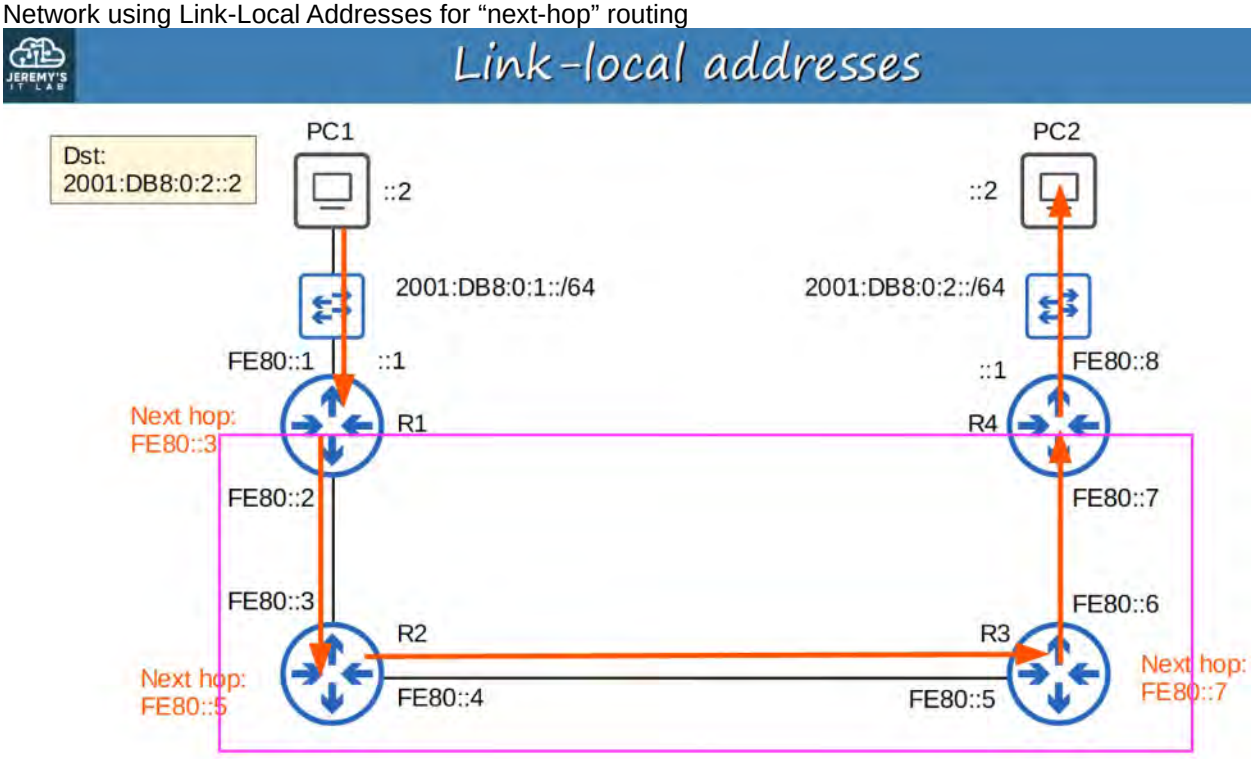
3. LINK-LOCAL ADDRESSES

- **Link-Local** IPv6 ADDRESSES are AUTOMATICALLY generated on IPv6-enabled INTERFACES
- Use command R1(config-if)# ipv6 enable on an interface to enable IPv6 on an INTERFACE

💡 Uses the ADDRESS block FE80::/10 (FE80:: to FEBF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF)

- The STANDARD states that the 54-bits AFTER FE80/10 should be ALL 0's so you won't see Link-Local ADDRESSES beginning with FE9, FEA, or FEB - ONLY FE8(!)
- The INTERFACE ID is generated using EUI-64 rules
- Link-Local means that these addresses are used for communication within a single link (SUBNET)
    - ROUTER will not route PACKETS with a Link-Local DESTINATION IPv6 ADDRESS
- Common uses of Link-Local Addresses:
    - Routing Protocol Peerings (OSPFv3 uses Link-Local Addresses for Neighbour Adjacencies)
    - NEXT-HOP ADDRESS for STATIC ROUTES
    - Neighbor Discovery Protocol (NDP, IPv6's replacement for ARP) uses Link-Local ADDRESSES to function

Network using Link-Local Addresses for "next-hop" routing



---

4. MULTICAST ADDRESSES

- **Unicast Addresses** are one-to-one
    - ONE SOURCE to ONE DESTINATION
- ***Broadcast*** Addresses are one-to-all
    - ONE SOURCE to ALL DESTINATIONS (within the subnet)
- **Multicast** Addresses are one-to-many
    - ONE SOURCE to MULTIPLE DESTINATIONS (that have joined the specific ***multicast*** group)

💡 IPv6 uses range FF00::/8 for multicast (FF00:: to FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF)

- **IPv6 doesn't use Broadcast** (there IS NO "Broadcast Address" in IPv6!)

YOU MUST KNOW THE MULTICAST ADDRESS FOR EACH ROUTER TYPE
NOTE that the IPv6 and IPv4 Addresses share the same last digit

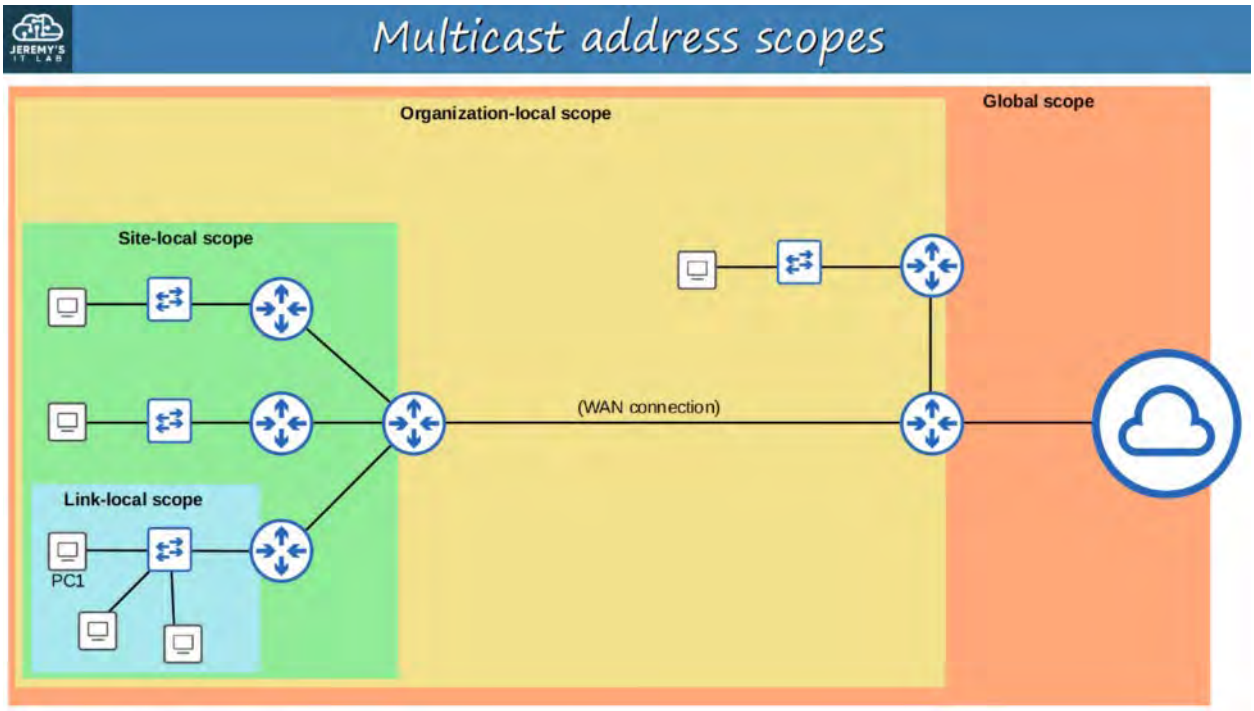
Purpose	IPv6 Address	IPv4 Address
All nodes/hosts (functions like broadcast)	FF02::1	224.0.0.1
All routers	FF02::2	224.0.0.2
All OSPF routers	FF02::5	224.0.0.5
All OSPF DRs/BDRs	FF02::6	224.0.0.6
All RIP routers	FF02::9	224.0.0.9
All EIGRP routers	FF02::A	224.0.0.10

MULTICAST ADDRESS SCOPES

- IPv6 defines multiple MULTICAST 'scopes' which indicate how far the PACKET should be forwarded
- The ADDRESS in the previous slide all use the 'link-local' scope (FF02), which stays in the LOCAL SUBNET

IPv6 Multicast Scope Types:

- **Interface-Local (FF01)**
  - The PACKET doesn't leave the LOCAL device
  - Can be used to SEND traffic to a SERVICE within the LOCAL device
- **Link-Local (FF02)**
  - The PACKET remains in the LOCAL SUBNET
  - ROUTERS will not route the PACKET between SUBNETS
- **Site-Local (FF05)**
  - The PACKET can be forwarded by ROUTERS
  - Should be limited to a SINGLE PHYSICAL LOCATION (not forwarded over a WAN)
- **Organization-Local (FF08)**
  - Wider in scope than Site-Local (an entire company / ORGANIZATION)
- **Global (FF0E)**
  - No boundaries
  - Possible to be ROUTED over the INTERNET



5. ANYCAST ADDRESS

- **ANYCAST is a NEW feature of IPv6**
- ANYCAST is 'one-to-one-of-many'
- Multiple ROUTERS are configured with the SAME IPv6 ADDRESS
    - They use a ROUTING PROTOCOL to advertise the address
    - When HOSTS sends PACKETS to that DESTINATION ADDRESS, ROUTERS will forward it to the NEAREST ROUTER configured with THAT IP ADDRESS (based on ROUTING METRIC)
- There is NO SPECIFIC ADDRESS range for ANYCAST ADDRESSES.
    - Use a regular UNICAST (Global Unicast, Unique Local) and specify THAT as an ANYCAST ADDRESS
    - R1(config-if)# ipv6 address 2000:db8:1:1::99/128 anycast

Anycast address configuration

```
R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8:1:1::99/128 anycast
R1(config-if)#
R1(config-if)#do show ipv6 interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::EF8:22FF:FE36:8500
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8::EF8:22FF:FE36:8500, subnet is 2001:DB8::/64 [EUI]
    2001:DB8:1:1::99, subnet is 2001:DB8:1:1::99/128 [ANY]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:99
    FF02::1:FF36:8500
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
```

6. OTHER IPv6 ADDRESSES

- The :: Address = The *unspecified* IPv6 ADDRESS
  - Can be used when a DEVICE doesn't yet know its IPv6 ADDRESS
  - IPv6 DEFAULT ROUTES are configured to ::/0
  - IPv4 equivalent: 0.0.0.0
- The ::1 Address = The Loopback Address
  - Used to test the PROTOCOL STACK on the LOCAL DEVICE
  - Messages sent to THIS ADDRESS are processed within the LOCAL DEVICE but not SENT to other DEVICES
  - IPv4 equivalent : 127.0.0.0 /8 address range

33. IPv6 : PART 3

CORRECTION TO PRIOR LECTURES:
RFC Requirements for IPv6 Address Representation

- Leading 0s MUST be removed
  - This - 2001 : 0db8 : 0000 : 0001 : 0f2a : 4fff : fea3 : 00b1
  - Becomes - 2001 : db8 : 0 : 1 : f2a : 4fff : fea3 : b1
- :: MUST be used to shorten the longest string of all-0 quartets
  - If there is only ONE all-0 quartet, don't use '::'
  - This - 2001 : 0000 : 0000 : 0000 : 0f2a : 0000 : 0000 : 00b1
  - Becomes - 2001 :: f2a : 0 : 0 : b1
- If there are two equal-length choices for the :: , use :: to the shorten the one on the LEFT
  - This - 2001 : 0db8 : 0000 : 0000 : 0f2a : 0000 : 0000 : 00b1
  - Becomes - 2001 : db8 :: f2a : 0 : 0 : b1
- Hexadecimal characters 'a', 'b', 'c', 'd', 'e', and 'f' MUST be written using lower-case, NOT upper case A B C D E F

IPv6 HEADER

Fixed header format

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Traffic Class								Flow Label																			
4	32	Payload Length																Next Header								Hop Limit							
8	64	Source Address																															
12	96																																
16	128																																
20	160																																
24	192	Destination Address																															
28	224																																
32	256																																
36	288																																

Length is ALWAYS 40 bytes (Fixed Header)
Version (4 bits)

- Indicates version of IP used
- Fixed value of '6' (0b0110) to indicate IPv6

Traffic Class (8 bits)

- Used for QoS (Quality of Service) to indicate high-priority traffic
- Example: IP phone traffic, live video calls, etc.

Flow Label (20 bits)

- Identifies specific traffic "flows" (communication between Source and Destination)

Payload Length (16 bits)

- Indicates the LENGTH of the PAYLOAD (the encapsulation LAYER 4 SEGMENT) **in bytes**
- The length of the IPv6 header, itself, isn't included, because it's ALWAYS 40 bytes

Next Header (8 bits)

- Indicates the TYPE of the 'next header' (header of the encapsulated SEGMENT)
  - Example: TCP or UDP
- Same function as the IPv4 header's 'Protocol' field

Hop Limit (8 bits)

- Value in this field decrements by 1 every time a ROUTER forwards it. If it reaches '0', the PACKET is discarded (similar to IPv4 TTL field)

Source Address (128 bits)

- Packet's SOURCE address

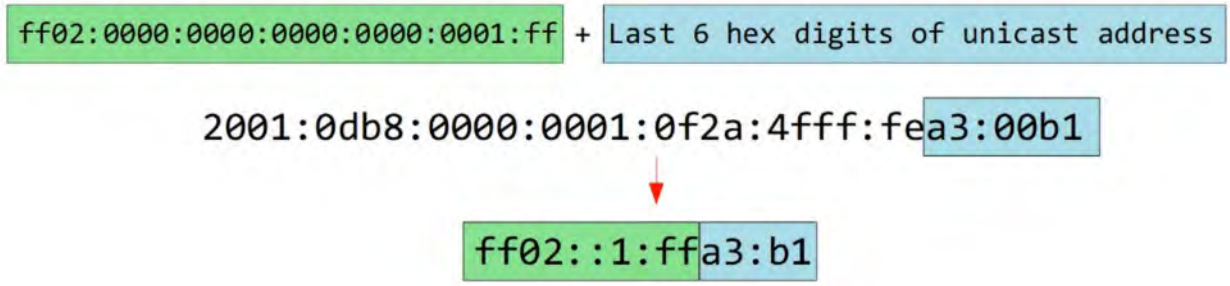
Destination Address (128 bits)

- Packet's DESTINATION address

SOLICITED-NODE MULTICAST ADDRESS

- An IPv6 SOLICITED-NODE Multicast Address is calculated from a UNICAST ADDRESS

How to generate a SOLICITED-NODE Multicast Address



Note the automatically joined group addresses for this IPv6 Interface

```
R1#sh ipv6 int g0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::EF8:22FF:FE36:8500
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8::EF8:22FF:FE36:8500, subnet is 2001:DB8::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF36:8500
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
```

NEIGHBOR DISCOVERY PROTOCOL (NDP)

- NEIGHBOR DISCOVERY PROTOCOL (NDP) is a PROTOCOL used with IPv6
- It has various functions and one of those functions is to replace ARP, which is no longer used in IPv6
- The ARP-like function of NDP uses ICMPv6 and SOLICITED-MODE Multicast Addresses to learn the MAC ADDRESS of other HOSTS (ARP in IPv4 uses Broadcast Messages)
- TWO MESSAGES types are used:
    - 1. NEIGHBOR SOLICITATION (NS)
        - ICMPv6 Type 135
    - 2. NEIGHBOR ADVERTISEMENT (NA)

- ICMPv6 Type 136

Neighbor Solicitation (NS)

- **Source IP**: R1 G0/0 IP
- **Destination IP**: R2 solicited-node multicast address
- **Source MAC**: R1 G0/0 MAC
- **Destination MAC**: Multicast MAC based on R2's solicited-node address

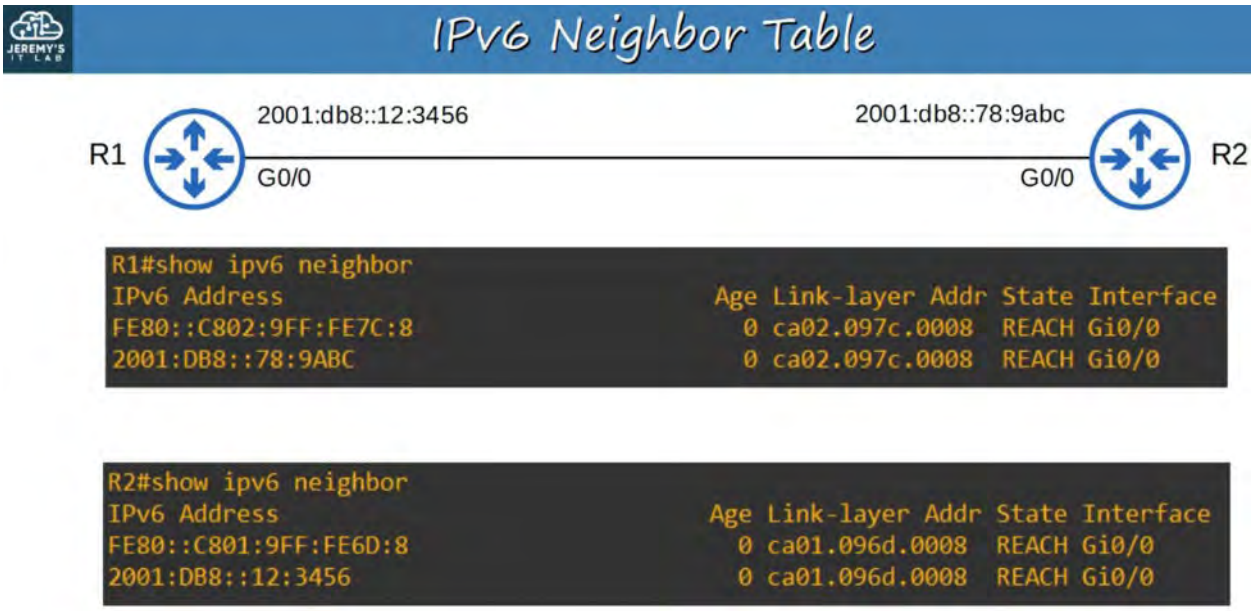
```
> Frame 6: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:01:09:6d:00:08 (ca:01:09:6d:00:08), Dst: IPv6mcast_ff:78:9a:bc (33:33:ff:78:9a:bc)
> Internet Protocol Version 6, Src: 2001:db8::12:3456, Dst: ff02::1:ff78:9abc
> Internet Control Message Protocol v6
```

Neighbor Advertisement (NA)

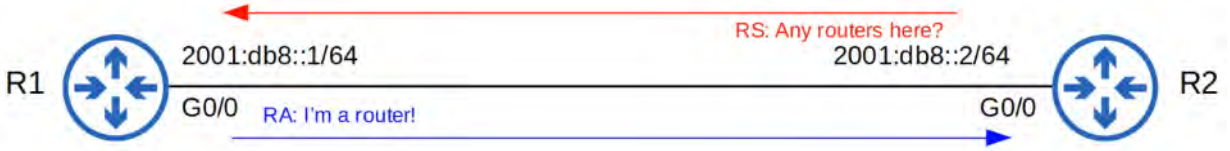
- **Source IP**: R2 G0/0 IP
- **Destination IP**: R1 G0/0 IP
- **Source MAC**: R2 G0/0 MAC
- **Destination MAC**: R1 G0/0 MAC

```
> Frame 7: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:02:09:7c:00:08 (ca:02:09:7c:00:08), Dst: ca:01:09:6d:00:08 (ca:01:09:6d:00:08)
> Internet Protocol Version 6, Src: 2001:db8::78:9abc, Dst: 2001:db8::12:3456
> Internet Control Message Protocol v6
```

IPv6 NEIGHBOR TABLE



- Another function of NDP allows HOSTS to automatically discover ROUTERS on the LOCAL NETWORK
- TWO MESSAGES are used for this process:
    - ROUTER SOLICITATION (RS)
        - ICMPv6 Type 133
        - Sent to Multicast Address FF02::2 (All Routers)
        - Asks ALL ROUTERS on the Local Link to identify themselves
        - Sent when an INTERFACE is enabled / HOST is connected to the NETWORK
    - ROUTER ADVERTISEMENT (RA)
        - ICMPv6 Type 134
        - Sent to Multicast Address FF02::1 (All Nodes)
        - The ROUTER announces its presence, as well as other information about the link
        - These messages are sent in response to RS messages
        - They are also sent periodically, even if the ROUTER hasn't received an RS



SLAAC

- Stands for **STATELESS ADDRESS AUTO-CONFIGURATION**
- HOSTS use the RS / RA messages to learn the IPv6 Prefix of the LOCAL LINK (ie: 2000:db8:: /64) and then automatically generate an IPv6 Address
- Using the ipv6 address prefix / prefix-length eui-64 command, you need to manually enter the prefix
- Using the ipv6 address autoconfig command, you DON'T need to enter the prefix. The device uses NDP to learn the prefix used on the local link
- The device will use EUI-64 to generate the INTERFACE ID or it will be randomly generated (depending on the device / maker)

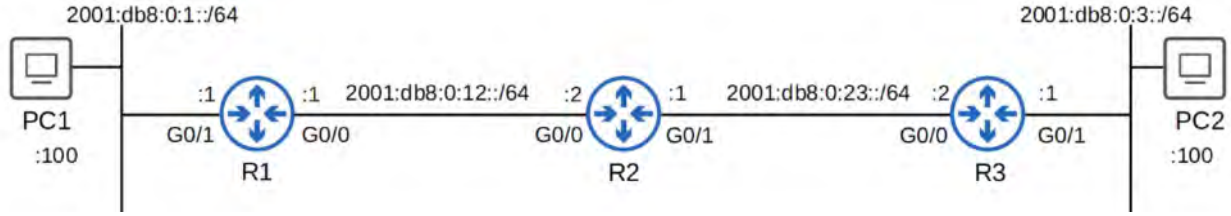
```
R2(config)#int g0/0
R2(config-if)#ipv6 address autoconfig
R2(config-if)#do show ipv6 interface brief
GigabitEthernet0/0     [up/up]
    FE80::EF8:22FF:FE56:A600
    2001:DB8::EF8:22FF:FE56:A600
GigabitEthernet0/1     [administratively down/down]
    unassigned
GigabitEthernet0/2     [administratively down/down]
    unassigned
GigabitEthernet0/3     [administratively down/down]
    unassigned
```

DUPLICATE ADDRESS DETECTION (DAD)

- One final point about NDP!
- Duplicate Address Detection (DAD) allows HOSTS to check if other devices on the Local Link are using the same IPv6 Address
- Any time an IPv6-enabled interface initializes (no shutdown command) or an IPv6 ADDRESS is configured on an INTERFACE (by any method: manual, SLAAC, etc.) it performs DAD
- DAD uses TWO MESSAGES you learned earlier : NS and NA
- The HOST will send an NS to its own IPv6 ADDRESS.
    - If it doesn't get a reply, it KNOWS the ADDRESS is unique
    - If it DOES get a reply, it means ANOTHER HOST on the NETWORK is already using that ADDRESS

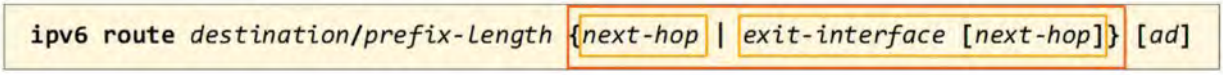
IPv6 STATIC ROUTING

- IPv6 ROUTING works the same as IPv4 ROUTING
- However, the TWO processes are separate on the ROUTER, and the TWO routing tables are separate, as well.
- IPv4 ROUTING is enabled BY DEFAULT
- IPv6 ROUTING is disabled BY DEFAULT
    - MUST BE ENABLED with the ipv6 unicast-routing command
- If IPv6 ROUTING is disabled, the ROUTER will be able to SEND and RECEIVE IPv6 traffic, but will not *route* IPv6 traffic (ie: will NOT FORWARD it between NETWORKS)



```
R1#show ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid
       lA - LISP away, a - Application
C    2001:DB8:0:1::/64 [0/0]
     via GigabitEthernet0/1, directly connected
L    2001:DB8:0:1::1/128 [0/0]
     via GigabitEthernet0/1, receive
C    2001:DB8:0:12::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L    2001:DB8:0:12::1/128 [0/0]
     via GigabitEthernet0/0, receive
L    FF00::/8 [0/0]
     via Null0, receive
```

- A CONNECTED NETWORK ROUTE is automatically added for EACH CONNECTED NETWORK
- A LOCAL HOST ROUTE is automatically added for each ADDRESS configured on the ROUTER
- Routes for Link-Local ADDRESSES are not added to the ROUTING TABLE



Everything is configured similar to normal static routes in IPv4
[AD] = Administrative Distance. You NEED this value in order to configure a STATIC ROUTE
DIRECTLY ATTACHED Static Route:

- Only the EXIT INTERFACE is specified
- ipv6 route destination / prefix-length exit-interface
- Example : ~~R1(config)# ipv6 route 2001:db8:0:3:: /64 g0/0~~

💡 In IPv6, you CANNOT use DIRECTLY ATTACHED Static Routes if the INTERFACE is an ETHERNET INTERFACE
RECURSIVE Static Route:

- Only the Next-Hop is specified
- ipv6 route destination / prefix-length next-hop
- Example: R1(config)# ipv6 route 2001:db8:0:3::/64 2001:db8:0:12::2

FULLY SPECIFIED Static Route:

- Both the Exit Interface and Next Hop are specified
- ipv6 route destination / prefix-length exit-interface next-hop
- Example: R1(config)# ipv6 route 2001:db8:0:3::/64 g0/0 2001:db8:0:12::2

(NOTE THAT THESE ROUTES ARE ALL RECURSIVE : They specify the Next-Hop)
NETWORK ROUTE:
R1(config)# ipv6 route 2001:db8:0::/64 2001:db8:0:12::2
This is a route to R3/PC2 NETWORK via R2's G0/0 INTERFACE
(We did this in Day 32's Lab)
HOST ROUTE:
R2(config)# ipv6 route 2001:db8:0:1::100/128 2001:db8:0:12::1

R2(config)# ipv6 route 2001:db8:0:3::100/128 2001:db8:0:23::2
This is a route from R2 to PC1 and PC2 using the “next hop” ADDRESSES of R1 and R3 G0/0 INTERFACES
Note the /128 prefix. This is how SPECIFIC IPv6 ADDRESSES are written
DEFAULT ROUTE:
R3(config)# ipv6 route ::/0 2001:db8:0:23::1
::/0 is the IPv6 equivalent of 0.0.0.0/0 in IPv4
FLOATING STATIC ROUTES:

- Require you to increase the [AD] number HIGHER than the currently used NETWORK IGP AD value

LINK-LOCAL NEXT HOPS:

```
R1(config)#ipv6 route 2001:db8:0:3::/64 FE80::EF8:22FF:FEE6:D300
% Interface has to be specified for a link-local nexthop
R1(config)#
R1(config)#ipv6 route 2001:db8:0:3::/64 g0/0 FE80::EF8:22FF:FEE6:D300
R1(config)#do show ipv6 route
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid
       lA - LISP away, a - Application
C    2001:DB8:0:1::/64 [0/0]
     via GigabitEthernet0/1, directly connected
L    2001:DB8:0:1::1/128 [0/0]
     via GigabitEthernet0/1, receive
S    2001:DB8:0:3::/64 [1/0]
     via FE80::EF8:22FF:FEE6:D300, GigabitEthernet0/0
C    2001:DB8:0:12::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L    2001:DB8:0:12::1/128 [0/0]
     via GigabitEthernet0/0, receive
L    FF00::/8 [0/0]
     via Null0, receive
```

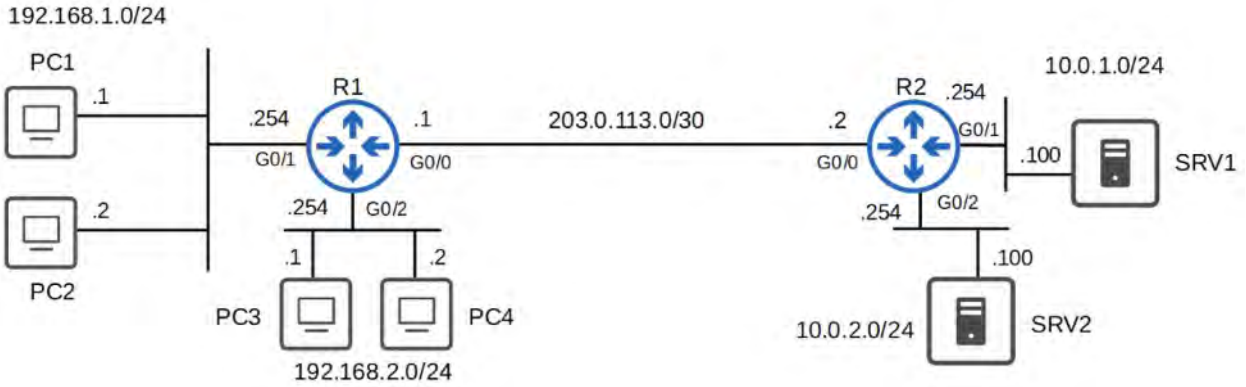
You HAVE to specify the INTERFACE name when using Link-Local Next-Hops
This is EXACTLY like a FULLY-SPECIFIED STATIC ROUTE

34. STANDARD ACCESS CONTROL LISTS (ACL)

WHAT ARE ACLs

- ACLs (Access Control Lists) have multiple uses
- In DAY 34 and DAY 35, we will focus on ACL's from a security perspective
- ACLs function as a "packet filter" - instructing the ROUTER to ALLOW or DENY specific traffic
- ACLs can filter traffic based on:
    - SOURCE / DESTINATION IP ADDRESSES
    - SOURCE / DESTINATION LAYER 4 PORTS
    - etc.

HOW ACLs WORK



💡 REQUIREMENTS:

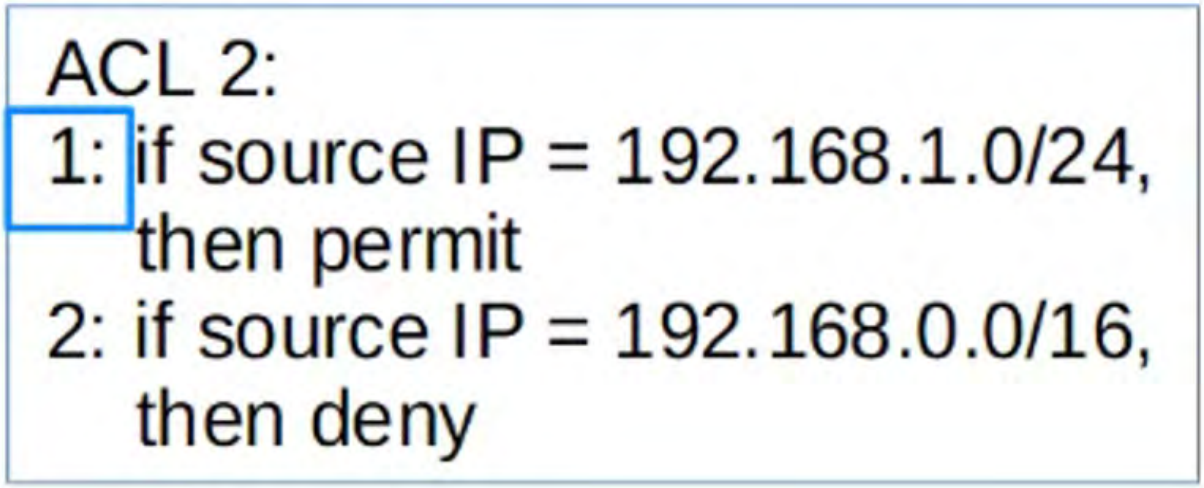
- Hosts in 192.168.1.0/24 should have ACCESS to the 10.0.1.0/24 NETWORK
- Hosts in 192.168.2.0/24 should not have ACCESS to the 10.0.10/24 NETWORK

ACLs are configured GLOBALLY on the ROUTER (Global Config Mode)

- They are an ordered sequence of ACEs (Access Control Entries)

```
ACL 1:
1: if source IP = 192.168.1.0/24,
   then permit
2: if source IP = 192.168.2.0/24,
   then deny
3: if source IP = any, then permit
```

- 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
- When a 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



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

IMPLICIT DENY

- What will happen if a PACKET doesn't match any of the entries in an ACL ?
- There is an INPLICIT DENY at the end of ALL ACL's
- The IMPLICIT DENY tells the ROUTER to DENY ALL TRAFFIC that doesn't match ANY of the configured entries in the ACL

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

- 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
  - o R1(config)# access-list *number* {deny | permit} *ip wildcard-mask*

This is an example of denying a SPECIFIC host's traffic

REMEMBER : 0.0.0.0 wildcard is the same as 255.255.255.255 or a /32 host

- Example : R1(config)# access-list 1 deny 1.1.1.1 0.0.0.0
- Example : R1(config)# access-list 1 deny 1.1.1.1(identical to the above)
- Example : R1(config)# access-list 1 deny host 1.1.1.1

If you want to permit ANY traffic from ANY source

- Example : R1(config)# access-list 1 permit any
- Example : R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255

If you want to make a description for a specific ACL

- Example : R1(config)# access-list 1 remark ## BLOCK BOB FROM ACCOUNTING ##

```
R1(config)#access-list 1 deny 1.1.1.1 0.0.0.0
R1(config)#access-list 1 permit 0.0.0.0 255.255.255.255
R1(config)#access-list 1 remark ## BLOCK BOB FROM ACCOUNTING ##
R1(config)#
R1(config)#do show access-lists
Standard IP access list 1
    10 deny   1.1.1.1
    20 permit any
R1(config)#
R1(config)#do show ip access-lists
Standard IP access list 1
    10 deny   1.1.1.1
    20 permit any
R1(config)#
R1(config)#do show running-config | include access-list
access-list 1 deny   1.1.1.1
access-list 1 permit any
access-list 1 remark ## BLOCK BOB FROM ACCOUNTING ##
R1(config)#
```

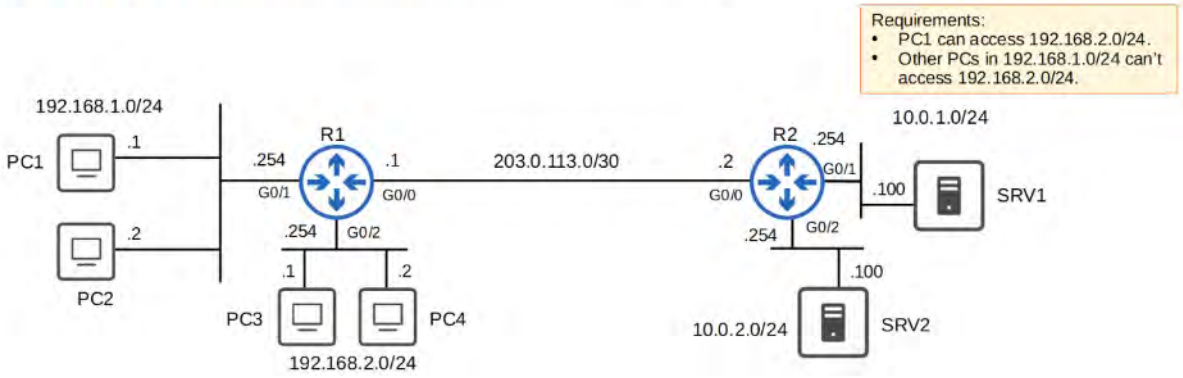
Order is important. Lower Numbers are processed FIRST

TO APPLY AN ACL TO AN INTERFACE

R1(config-if)# ip access-group *number* {in | out}

Standard Numbered ACLs

```
R1(config)#access-list 1 permit 192.168.1.1
R1(config)#access-list 1 deny 192.168.1.0 0.0.0.255
R1(config)#access-list 1 permit any
R1(config)#
R1(config)#interface g0/2
R1(config-if)#ip access-group 1 out
R1(config-if)#
```



WHY WAS THIS RULE PLACED ON G0/2 OUT ?

💡 STANDARD ACLs should be applied as CLOSE to the DESTINATION as possible!

STANDARD NAMED ACLs

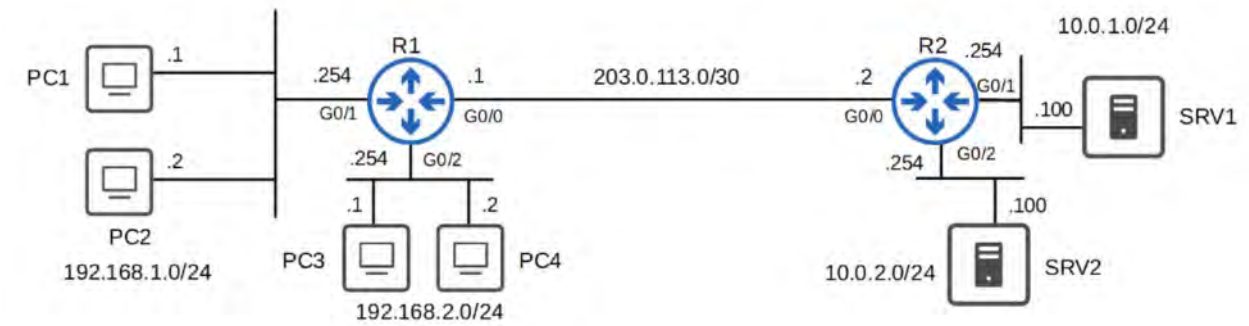
- Standard ACLs match traffic based only on the SOURCE IP ADDRESS of the PACKET
- NAMED ACLs are identified with a NAME (ie: 'BLOCK_BOB')
- STANDARD NAMED ACLs are configured by entering 'standard named ACL config mode' then configuring EACH entry within that config mode
    - o R1(config)# ip access-list standard *acl-name*
    - o 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
```

```
R1#show access-lists
Standard IP access list BLOCK_BOB
    5 deny   1.1.1.1
    10 permit any
R1#
R1#show running-config | section access-list
ip access-list standard BLOCK_BOB
 deny   1.1.1.1
 permit any
 remark ## CONFIGURED NOV 21 2020 ##
R1#
```

Requirements:

- PCs in 192.168.1.0/24 can't access 10.0.2.0/24.
- PC3 can't access 10.0.1.0/24.
- Other PCs in 192.168.2.0/24 can access 10.0.1.0/24.
- PC1 can access 10.0.1.0/24.
- Other PCs in 192.168.1.0/24 can't access 10.0.1.0/24.



Here are the configurations for the above:

```
R2(config)#ip access-list standard TO_10.0.2.0/24
R2(config-std-nacl)#deny 192.168.1.0 0.0.0.255
R2(config-std-nacl)#permit any
R2(config-std-nacl)#interface g0/2
R2(config-if)#ip access-group TO_10.0.2.0/24 out
R2(config-if)#
R2(config-if)#ip access-list standard TO_10.0.1.0/24
R2(config-std-nacl)#deny 192.168.2.1
R2(config-std-nacl)#permit 192.168.2.0 0.0.0.255
R2(config-std-nacl)#permit 192.168.1.1
R2(config-std-nacl)#deny 192.168.1.0 0.0.0.255
R2(config-std-nacl)#permit any
R2(config-std-nacl)#interface g0/1
R2(config-if)#ip access-group TO_10.0.1.0/24 out
R2(config-if)#
```

Note, however, how the order is when viewing the ACLs

```
R2#show ip access-lists
Standard IP access list TO_10.0.1.0/24
    30 permit 192.168.1.1
    10 deny   192.168.2.1
    20 permit 192.168.2.0, wildcard bits 0.0.0.255
    40 deny   192.168.1.0, wildcard bits 0.0.0.255
    50 permit any
Standard IP access list TO_10.0.2.0/24
    10 deny   192.168.1.0, wildcard bits 0.0.0.255
    20 permit any
R2#
```

WHY THE REORDERING?

- The router may re-order the /32 entries.
- This improves the efficiency of processing the ACL.
- It **does not** change the effect of the ACL.
- This applies to both standard named and standard numbered ACLs.
- Packet Tracer does not do this.

CISCOs PACKET TRACER does not reorder these, however.

35. EXTENDED ACCESS CONTROL LISTS (EACL)

ANOTHER WAY TO CONFIGURE NUMBERED ACLs

- 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
```

Configuring numbered ACLs with subcommands

```
R1(config)#ip access-list standard ?
  <1-99>       Standard IP access-list number
  <1300-1999>  Standard IP access-list number (expanded range)
  WORD         Access-list name

R1(config)#ip access-list standard 1
R1(config-std-nacl)#deny 192.168.1.1
R1(config-std-nacl)#permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 permit any
R1(config-std-nacl)#
```

ADVANTAGES OF NAMED ACL CONFIG MODE

- You can easily DELETE individual entries in the ACL with NO *entry-number*
- You can easily DELETE individual entries in the ACL with NO *sequence-number*

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.3.0, wildcard bits 0.0.0.255
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#no 30
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    40 permit any
R1(config-std-nacl)#
```

This doesn't work with NUMBERED access lists

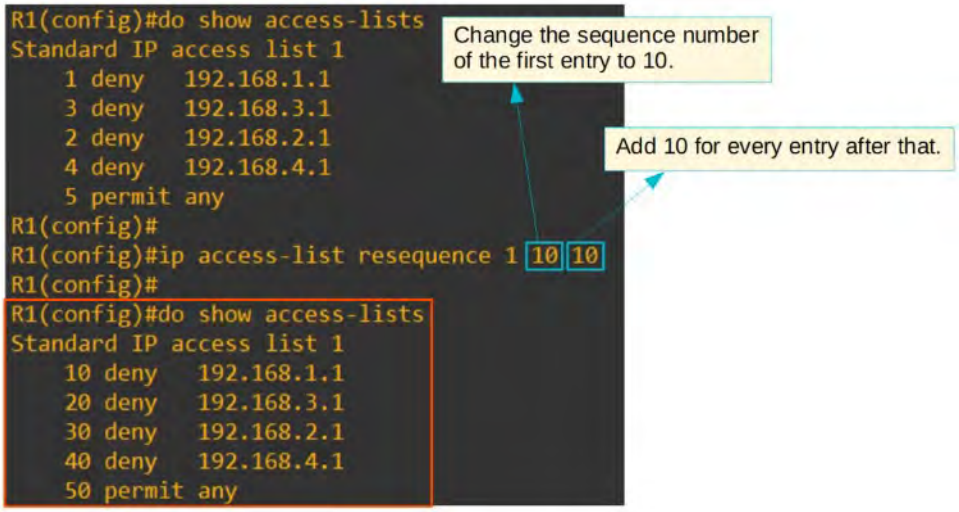
```
R1(config)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.3.0, wildcard bits 0.0.0.255
    40 permit any
R1(config)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 deny   192.168.1.2
access-list 1 deny   192.168.3.0 0.0.0.255
access-list 1 permit any
R1(config)#no access-list 1 deny 192.168.3.0 0.0.0.255
R1(config)#do show access-lists
R1(config)#do show running-config | section access-list
R1(config)#
```

- You can insert NEW entries in-between other entries by specifying the SEQUENCE NUMBER

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#30 deny 192.168.2.0 0.0.0.255
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.2.0, wildcard bits 0.0.0.255
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 deny   192.168.1.2
access-list 1 deny   192.168.2.0 0.0.0.255
access-list 1 permit any
```

RESEQUENCING ACLs

- There is a *resequencing* function that helps edit ACLs
- The command is R1(config)#ip access-list resequence *acl-id starting-seq-num increment*



EXTENDED NUMBERS AND NAMED 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**
- Processed from TOP to BOTTOM, just like STANDARD ACLs
- However, they 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**
    - D**estination Address**

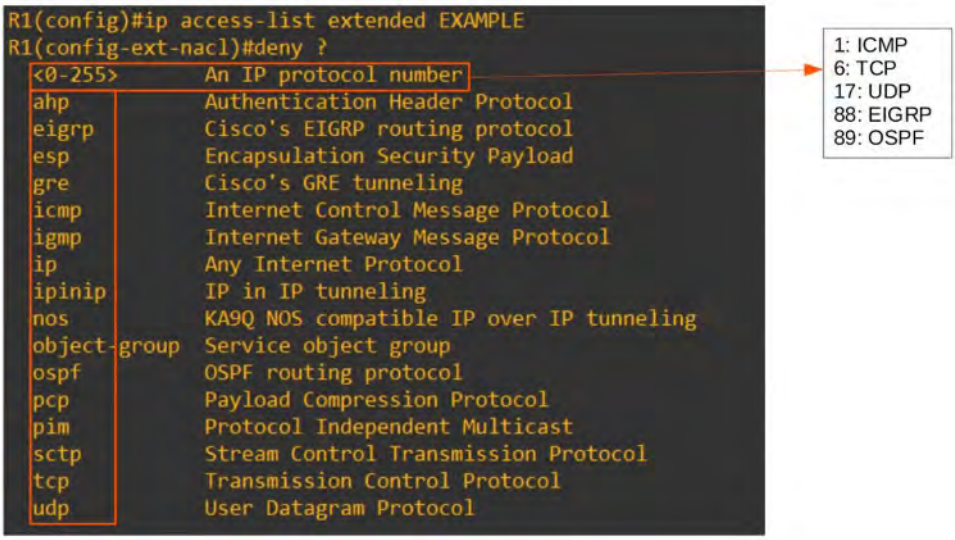
EXTENDED NUMBERED ACL

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

EXTENDED NAMED ACL

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

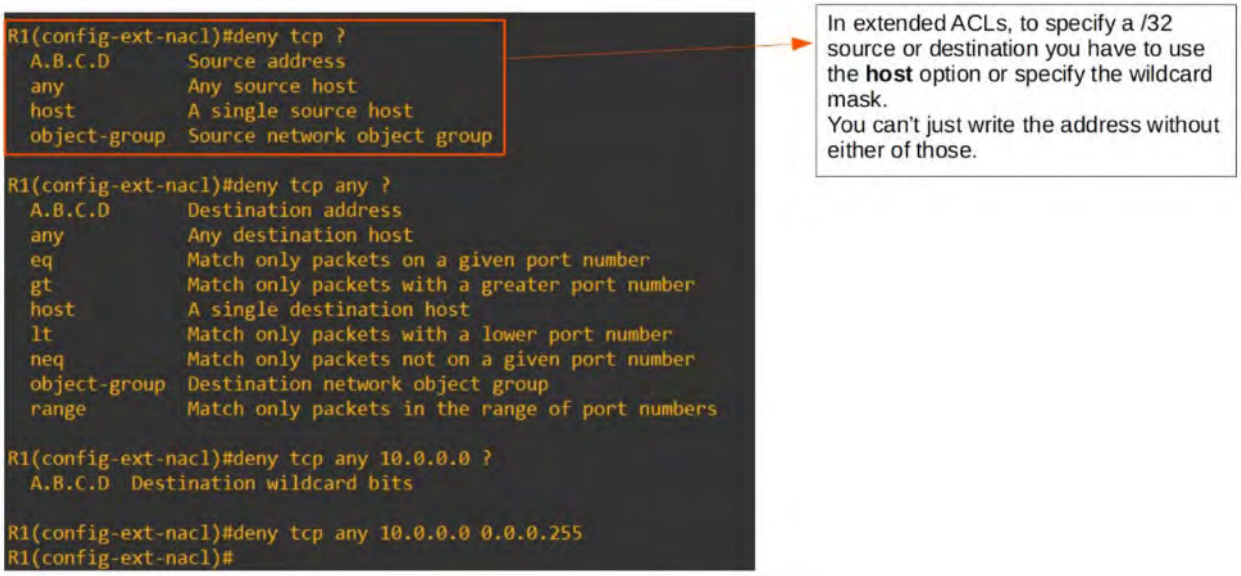
MATCHING THE PROTOCOL



IP Protocol Number is the number used in the IPv4 Header Protocol field

Examples: (1) ICMP, (6) TCP, (17) UDP, (88) EIGRP, (89) OSPF

MATCHING THE SOURCE / DESTINATION IP ADDRESS



This command:

💡 `R1(config-ext-nacl)#deny tcp any 10.0.0.0 0.0.0.255`

Deny ALL PACKETS that encapsulate a TCP segment from ANY source to DESTINATION 10.0.0.0/24

PRACTICE QUESTIONS:

1. ALLOW ALL TRAFFIC

R1(config-ext-nacl)# permit ip any any (ip is used for "all protocols")

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)

eq = equal than
gt = greater than
lt = less than
neq = not equal to
range = range of ports
You can use either the PORT NUMBER or the specific TYPE (that has a KNOWN PORT NUMBER)

```
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

If you specify the protocol, source IP, source port, destination IP, destination port, etc, a packet must match ALL of those values to match the ACL entry. Even if it matches all except one of the parameters, the packet won't match that entry of the ACL.

PRACTICE QUESTIONS 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 host 2.2.2.2 eq 443

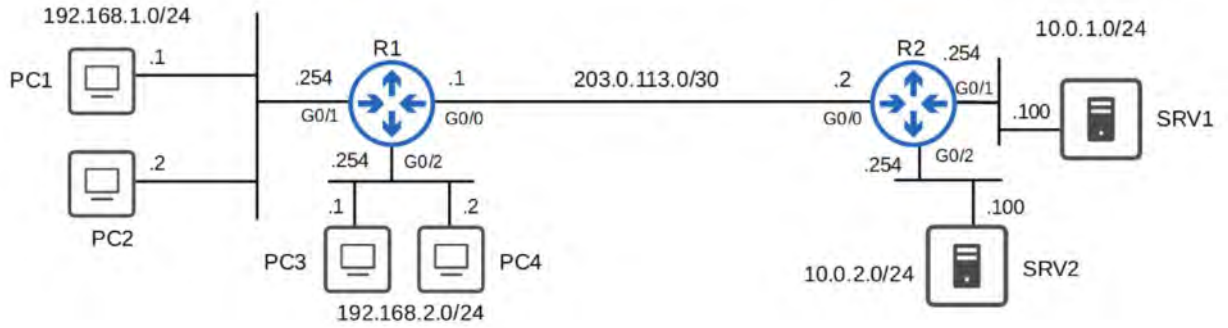
2. PREVENT ALL HOSTS using SOURCE UDP Port Numbers from 20000 to 30000 from accessing the server at 3.3.3.3/32

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

EXAMPLE NETWORK



Extended ACLs should be applied as close to the source as possible, to limit how far the packets travel in the network before being denied.

(Standard ACLs are less specific, so if they are applied close to the source there is a risk of blocking more traffic than intended)

REQUIREMENTS:

- Hosts in 192.168.1.0/24 can't use HTTPS to access SRV1
- Hosts in 192.168.2.0/24 can't access 10.0.2.0/24
- NONE of the hosts in 192.168.1.0/24 or 192.168.2.0/24 can ping 10.0.1.0/24 OR 10.0.2.0/24

EXTENDED ACL #1 (Applied at R1 G0/1 INBOUND interface)

R1(config)# ip access-list extended HTTP_SRV1 R1(config-ext-nacl)# deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443

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

R1(config-ext-nacl)# int g0/1

R1(config-if)# ip access-group HTTP_SRV1 in

EXTENDED ACL #2 (APPLIED at R1 G0/2 INBOUND interface)

R1(config)# ip access-list extended BLOCK_10.0.2.0

R1(config-ext-nacl)# deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255

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

R1(config-ext-nacl)# int g0/2

R1(config-if)# ip access-group BLOCK_10.0.2.0 in

EXTENDED ACL #3 (APPLIED at R1 g0/0 OUTBOUND interface)

R1(config)# ip access-list extended BLOCK_ICMP

R1(config-ext-nacl)# deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255

R1(config-ext-nacl)# deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255

R1(config-ext-nacl)# deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255

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

R1(config-ext-nacl)# int g0/0

R1(config-if)# ip access-group BLOCK_ICMP out

What the EXTENDED ACLs look like

```
R1#show access-lists
Extended IP access list BLOCK_10.0.2.0/24
    10 deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255
    20 permit ip any any
Extended IP access list BLOCK_ICMP
    10 deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
    20 deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255
    30 deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255
    40 permit ip any any
Extended IP access list HTTP_SRV1
    10 deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443
    20 permit ip any any
```

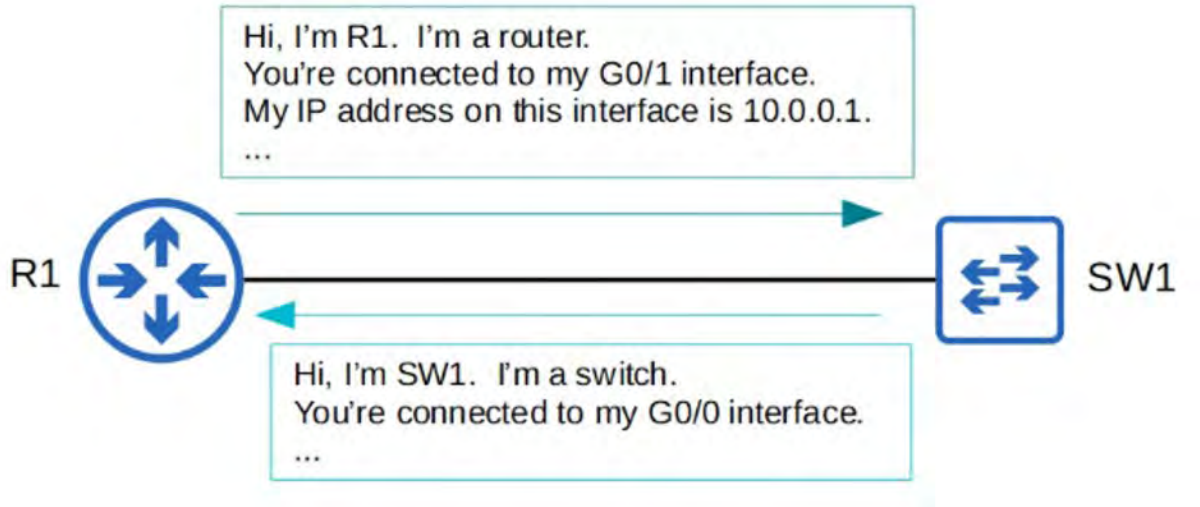
HOW TO SEE WHICH EXTENDED ACL's ARE APPLIED TO AN INTERFACE

```
R1#show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 203.0.113.1/30
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is BLOCK_ICMP
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
```

36. CDP and LLDP (Layer 2 Discovery Protocol)

INTRO TO LAYER 2 DISCOVERY PROTOCOLS

- LAYER 2 DISCOVERY PROTOCOL, such as CDP and LLDP share information WITH and DISCOVER information about NEIGHBORING (Connected) DEVICES
- The SHARED INFORMATION includes:
    - Hostname
    - IP Address
    - Device Type
    - etcetera.
- **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.

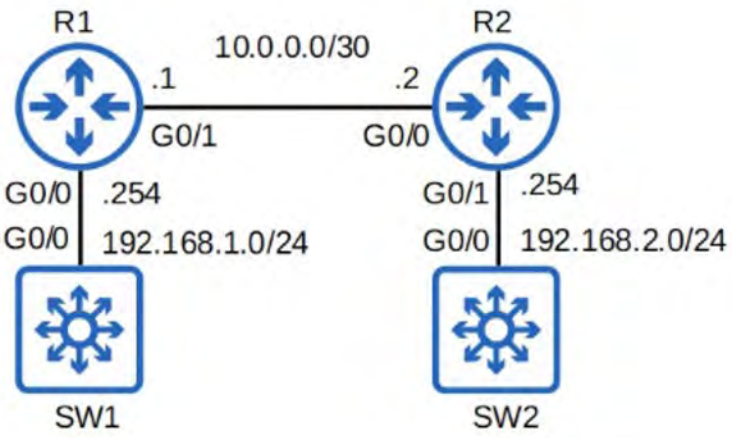


CISCO DISCOVERY PROTOCOL (CDP)

- 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 hold-time 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




```
R1#show cdp
Global CDP information:
        Sending CDP packets every 60 seconds
        Sending a holdtime value of 180 seconds
        Sending CDPv2 advertisements is  enabled
R1#
R1#show cdp traffic
CDP counters :
        Total packets output: 105, Input: 112
        Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
        No memory: 0, Invalid packet: 0,
        CDP version 1 advertisements output: 0, Input: 0
        CDP version 2 advertisements output: 105, Input: 112
R1#
```

```
R1#show cdp interface
GigabitEthernet0/0 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/2 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/3 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds

 cdp enabled interfaces : 4
 interfaces up          : 2
 interfaces down        : 2
```

CDP NEIGHBOR TABLES

```
R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW1              Gig 0/0           153           R S I           Gig 0/0
R2               Gig 0/1           146            R B            Gig 0/0

Total cdp entries displayed : 2
R1#
```

“Device ID” = What devices were DISCOVERED by CDP
“Local Intrface” = What LOCAL device interface the neighbors are connected to
“Holdtime” = Hold-time countdown in seconds (0 = device removed from table)
“Capabilities” = Refers to Capability Codes table (located above output)
“Platform” = Displays the MODEL of the Neighbor Device
“Port ID” = Neighbor ports that LOCAL device is connected to

MORE DETAILED OUTPUT

```
R1#show cdp neighbors detail
-------------------------
Device ID: SW1
Entry address(es):
Platform: Cisco ,  Capabilities: Router Switch IGMP
Interface: GigabitEthernet0/0,  Port ID (outgoing port): GigabitEthernet0/0
Holdtime : 174 sec

Version :
Cisco IOS Software, vios_l2 Software (vios_l2-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST ENGINEERING ESTG_WEEKLY BUILD, synced to  END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Tue 28-Jul-15 18:52 by sasyamal

advertisement version: 2
VTP Management Domain: ''
Native VLAN: 1
Duplex: full

-------------------------
Device ID: R2
Entry address(es):
  IP address: 10.0.0.2
Platform: Cisco ,  Capabilities: Router Source-Route-Bridge
Interface: GigabitEthernet0/1,  Port ID (outgoing port): GigabitEthernet0/0
Holdtime : 163 sec

Version :
Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(2)T, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Tue 22-Mar-16 16:19 by prod_rel_team

advertisement version: 2
Duplex: full
Management address(es):
  IP address: 10.0.0.2


Total cdp entries displayed : 2
```

“Version” = shows what version of Cisco’s IOS is running on the device

---

SHOW SPECIFIC CDP NEIGHBOR ENTRY

```
R1#show cdp entry R2
-------------------------
Device ID: R2
Entry address(es):
  IP address: 10.0.0.2
Platform: Cisco ,  Capabilities: Router Source-Route-Bridge
Interface: GigabitEthernet0/1,  Port ID (outgoing port): GigabitEthernet0/0
Holdtime : 178 sec

Version :
Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(2)T, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Tue 22-Mar-16 16:19 by prod_rel_team

advertisement version: 2
Duplex: full
Management address(es):
  IP address: 10.0.0.2
```

CDP CONFIGURATION COMMANDS

- 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 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 time *seconds*
- Configure the CDP holdtime: R1(config)# cdp holdtime *seconds*
- ENABLE / DISABLE CDPv2: R1(config)# [no] cdp advertise-v2

LINK LAYER DISCOVERY PROTOCOL (LLDP)

- 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`

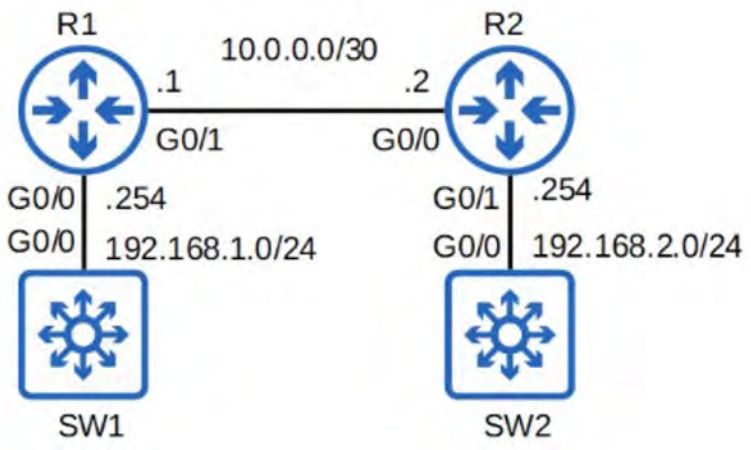
- When a DEVICE receives an LLDP message, it PROCESSES and DISCARDS the message. It does NOT forward it to OTHER DEVICES
- By DEFAULT, LLDP Messages are sent once every **30 seconds**
- By DEFAULT, LLDP Holdtime is **120 seconds**
- LLDP has an additional timer called the 'reinitialization delay'
    - 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 INTERFACES (rx): R1(config-if)# lldp receive

YOU NEED TO ENABLE BOTH TO SEND AND RECEIVE (Unless you want to only enable SEND or RECEIVE LLDP Messages)

- 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*



```
R1#show lldp traffic

LLDP traffic statistics:
    Total frames out: 4
    Total entries aged: 0
    Total frames in: 3
    Total frames received in error: 0
    Total frames discarded: 0
    Total TLVs discarded: 0
    Total TLVs unrecognized: 0
R1#
R1#show lldp interface

GigabitEthernet0/0:
    Tx: enabled
    Rx: enabled
    Tx state: IDLE
    Rx state: WAIT FOR FRAME

GigabitEthernet0/1:
    Tx: enabled
    Rx: enabled
    Tx state: IDLE
    Rx state: WAIT FOR FRAME

GigabitEthernet0/2:
    Tx: enabled
    Rx: enabled
    Tx state: INIT
    Rx state: WAIT PORT OPER

GigabitEthernet0/3:
    Tx: enabled
    Rx: enabled
    Tx state: INIT
    Rx state: WAIT PORT OPER
```

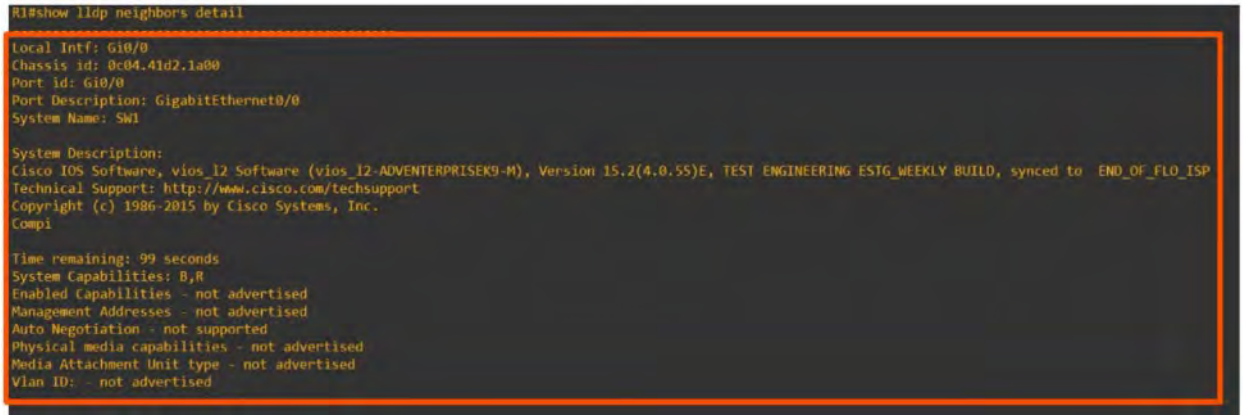
SHOW LLDP STATUS

```
R1#show lldp

Global LLDP Information:
    Status: ACTIVE
    LLDP advertisements are sent every 30 seconds
    LLDP hold time advertised is 120 seconds
    LLDP interface reinitialisation delay is 2 seconds
```

SHOW ALL LLDP NEIGHBORS

```
R1#show lldp neighbors
Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID           Local Intf     Hold-time  Capability      Port ID
SW1                 Gi0/0          120                        Gi0/0
R2                  Gi0/1          120        R               Gi0/0

Total entries displayed: 2
```

SHOW LLDP NEIGHBORS in DETAIL



SHOW SPECIFIC LLDP DEVICE ENTRY

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

37. NTP

WHY IS TIME IMPORTANT FOR NETWORK DEVICES?

- All DEVICES have an INTERNAL CLOCK (ROUTERS, SWITCHES, PCs, etc)
- In CISCO IOS, you can view the time with the show clock command

```
R1#show clock
*00:16:00.857 UTC Sat Dec 26 2020
```

The default time zone is UTC (Coordinated Universal Time).

- If you use the show clock detail command, you can see the TIME SOURCE

```
R1#show clock detail
*00:19:49.411 UTC Sat Dec 26 2020
Time source is hardware calendar
```

* = time is not considered authoritative

The hardware calendar is the default time source.

- The INTERNAL HARDWARE CLOCK of a DEVICE will “drift’ over time, so it’s NOT the ideal time source.
- From a CCNA perspective, the most important reason to have accurate time on a DEVICE is to have ACCURATE logs for troubleshooting
- **Syslog**, the protocol used to keep device logs, will be covered in a later video

Command: show logging

```
R2#show logging
!output abbreviated!
*Dec 27 00:50:20.005: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.122.192 on GigabitEthernet0/0 from LOADING to FULL,
Loading Done
*Dec 27 01:06:38.653: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done
*Dec 27 01:07:07.311: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done
*Dec 27 01:08:29.924: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from FULL to DOWN, Neighbor
Down: Dead timer expired
*Dec 27 01:09:10.714: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done

R2#show clock
*01:17:06.706 UTC Sun Dec 27 2020
```

Note : R3’s time stamp is completely different than R2’s !!!

```
R3#show logging
!output abbreviated!
May 23 16:24:17.320: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from LOADING to FULL, Loading
Done
May 23 16:25:08.758: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from FULL to DOWN, Neighbor
Down: Interface down or detached
May 23 16:25:10.714: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to down
May 23 16:25:11.716: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to down
May 23 16:26:14.976: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
May 23 16:26:15.977: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
May 23 16:26:20.618: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from LOADING to FULL, Loading
Done

R3#show clock
16:30:37.020 UTC Fri May 23 2008
```

MANUAL TIME CONFIGURATION

- You can manually configure the TIME on the DEVICE with the clock set command

```
R2#clock set ?
  hh:mm:ss  Current Time

R2#clock set 14:30:00 ?
  <1-31>  Day of the month
  MONTH   Month of the year

R2#clock set 14:30:00 27 ?
  MONTH  Month of the year

R2#clock set 14:30:00 27 Dec ?
  <1993-2035>  Year

R2#clock set 14:30:00 27 Dec 2020 ?
  <cr>

R2#clock set 14:30:00 27 Dec 2020
R2#show clock detail
14:30:05.887 UTC Sun Dec 27 2020
Time source is user configuration
```

- Although the HARDWARE CALENDAR (built-in clock) is the DEFAULT time-source, the HARDWARE CLOCK and SOFTWARE CLOCK are separate and can be configured separately.

HARDWARE CLOCK (CALENDAR) CONFIGURATION

- You can MANUALLY configure the HARDWARE CLOCK with the calendar set command

```
R2#calendar set 14:35:00 ?
  <1-31>  Day of the month
  MONTH   Month of the year

R2#calendar set 14:35:00 27 ?
  MONTH  Month of the year

R2#calendar set 14:35:00 27 Dec ?
  <1993-2035>  Year

R2#calendar set 14:35:00 27 Dec 2020 ?
  <cr>

R2#calendar set 14:35:00 27 Dec 2020
R2#show calendar
14:35:07 UTC Sun Dec 27 2020
```

- Typically, you will want to SYNCHRONIZE the ‘clock’ and ‘calendar’
- Use the command clock update-calendar to sync the calendar to the clock’s time
- Use the command clock read-calendar to sync the clock to the calendar’s time

```
R2#show clock
14:38:14.301 UTC Sun Dec 27 2020
R2#show calendar
00:00:03 UTC Sun Dec 27 2020
R2#clock update-calendar
R2#show clock
14:38:22.181 UTC Sun Dec 27 2020
R2#show calendar
14:38:23 UTC Sun Dec 27 2020
```

```
R2#show clock
00:00:15.788 UTC Mon Sep 6 1993
R2#show calendar
14:55:07 UTC Sun Dec 27 2020
R2#clock read-calendar
R2#show clock
14:55:12.522 UTC Sun Dec 27 2020
R2#show calendar
14:55:15 UTC Sun Dec 27 2020
```

CONFIGURING THE TIME ZONE

- You can configure the time zone with the clock timezone command

```
R2(config)#do show clock
15:13:33.985 UTC Sun Dec 27 2020
R2(config)#clock timezone ?
  WORD  name of time zone

R2(config)#clock timezone JST ?
  <-23 - 23>  Hours offset from UTC

R2(config)#clock timezone JST 9 ?
  <0-59>  Minutes offset from UTC
  <cr>

R2(config)#clock timezone JST 9
R2(config)#do show clock
00:13:45.414 JST Mon Dec 28 2020
R2(config)#do clock set 15:15:00 Dec 27 2020
R2(config)#do show clock
15:15:02.129 JST Sun Dec 27 2020
```

DAYLIGHT SAVING TIME (SUMMER TIME)

```
R2(config)#clock summer-time ?
  WORD  name of time zone in summer
R2(config)#clock summer-time EDT ?
  date       Configure absolute summer time
  recurring  Configure recurring summer time
R2(config)#clock summer-time EDT recurring ?
  <1-4>  Week number to start
  first  First week of the month
  last   Last week of the month
  <cr>
R2(config)#clock summer-time EDT recurring 2 ?
  DAY  Weekday to start
R2(config)#clock summer-time EDT recurring 2 Sunday ?
  MONTH  Month to start
R2(config)#clock summer-time EDT recurring 2 Sunday March ?
  hh:mm  Time to start (hh:mm)
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 ?
  <1-4>  Week number to end
  first  First week of the month
  last   Last week of the month
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 ?
  DAY  Weekday to end
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday ?
  MONTH  Month to end
R2(config)#$r-time EDT recurring 2 Sunday March 02:00 1 Sunday November ?
  hh:mm  Time to end (hh:mm)
R2(config)#$ recurring 2 Sunday March 02:00 1 Sunday November 02:00 ?
  <1-1440>  Offset to add in minutes
  <cr>
R2(config)#$ recurring 2 Sunday March 02:00 1 Sunday November 02:00
```

Canada	Northern America	Northern	Second Sunday March at 02:00 local standard time (for most of Canada)	First Sunday November at 02:00 local daylight saving time (for most of Canada)

Full command :

R1(config)# clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday November 02:00

This covers the START of Daylight Savings and the end of Daylight Savings

SUMMARY OF COMMANDS

```
R1# show clock

R1# show clock detail

R1# clock set hh:mm:ss {day|month} {month|day} year

R1# show calendar

R1# calendar set hh:mm:ss {day|month} {month|day} year


R1(config)# clock timezone name hours-offset [minutes-offset]

R1(config)# clock summer-time recurring name start end [offset]
```

NTP BASICS

- Manually configuring the time on DEVICES is NOT Scalable
- The manually configured clocks will “drift”, resulting in inaccurate time
- NTP (Network Time Protocol) allows AUTOMATIC synchronization of TIME over a NETWORK
- NTP CLIENTS request the TIME from NTP SERVERS
- A DEVICE can be an NTP SERVER and an NTP CLIENT at the same time
- NTP allows accuracy of TIME with ~1 millisecond if the NTP SERVER is in the same LAN - OR within ~50 milliseconds if connecting to the NTP SERVER over a WAN / the INTERNET
- Some NTP SERVERS are ‘better’ than others. The ‘distance’ of an NTP SERVER from the original **reference clock** is called **stratum**

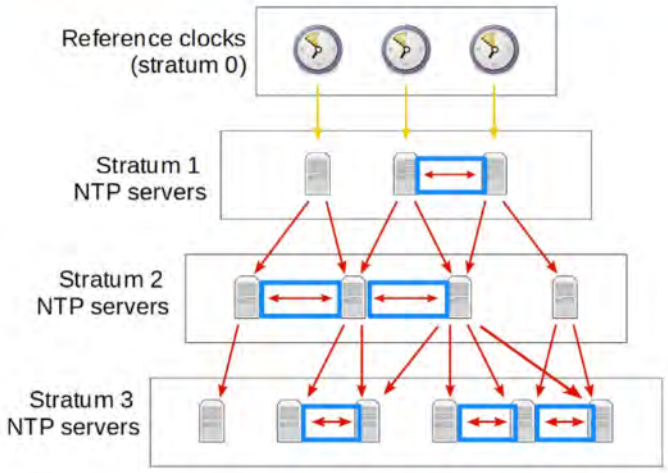
💡 NTP uses UDP port 123 to communicate

REFERENCE CLOCK

- A REFERENCE CLOCK is usually a VERY accurate time device like an ATOMIC CLOCK or GPS CLOCK
- REFERENCE CLOCKS are **stratum 0** within the NTP hierarchy
- NTP SERVERS directly connected to REFERENCE CLOCKS are **stratum 1**

NTP Hierarchy

- Reference clocks are **stratum 0**.
- **Stratum 1** NTP servers get their time from reference clocks.
- **Stratum 2** NTP servers get their time from stratum 1 NTP servers.
- **Stratum 3** NTP servers get their time from stratum 2 NTP servers.
- **Stratum 15** is the maximum. Anything above that is considered unreliable.
- Devices can also ‘peer’ with devices at the same stratum to provide more accurate time.

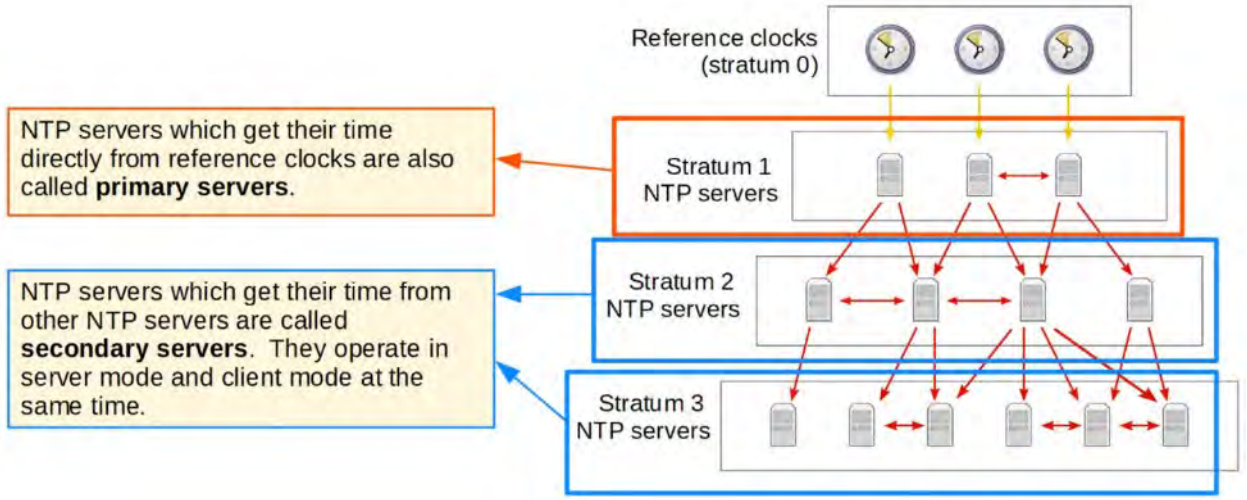


(Peering with Devices is called …)

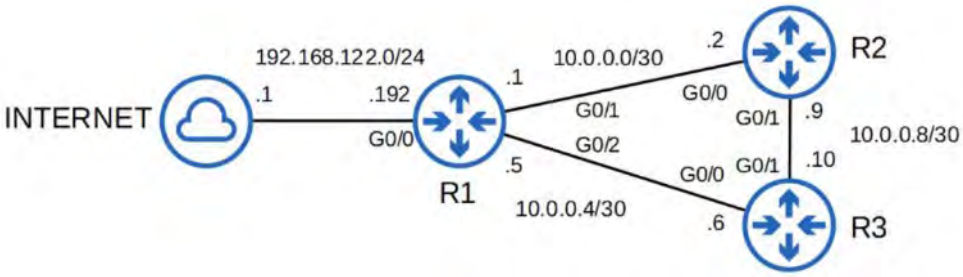
This is called 'symmetric active' mode.
Cisco devices can operate in three NTP modes:

- Server mode
- Client mode
- Symmetric active mode

- An NTP CLIENT can SYNC to MULTIPLE NTP SERVERS



NTP CONFIGURATION



```
C:\Users\user>nslookup time.google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    time.google.com
Addresses:  2001:4860:4806::
          2001:4860:4806:c::
          2001:4860:4806:8::
          2001:4860:4806:4::
          216.239.35.12
          216.239.35.8
          216.239.35.4
          216.239.35.0
```

```
R1(config)#ntp server 216.239.35.0 prefer
R1(config)#ntp server 216.239.35.4
R1(config)#ntp server 216.239.35.8
R1(config)#ntp server 216.239.35.12
```

Using key argument “prefer” makes a given server the PREFERRED SERVER
(To show configuration servers)

```
R1#show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~216.239.35.0    .GOOG.           1     43     64    17 62.007 1401.54  0.918
+~216.239.35.8    .GOOG.           1     43     64    17 64.220 1416.65  0.939
+~216.239.35.4    .GOOG.           1     47     64    17 57.669 1402.11  0.916
+~216.239.35.12   .GOOG.           1     39     64    17 62.229 1409.03  0.960
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

sys.peer = This is the SERVER that the current ROUTER (R1) is being synchronized to
st = Stratum Tier
(To show NTP Status)

```
R1#show ntp status
Clock is synchronized, stratum 2, reference is 216.239.35.12
nominal freq is 1000.0003 Hz, actual freq is 999.5003 Hz, precision is 2**14
ntp uptime is 295800 (1/100 of seconds), resolution is 1001
reference time is E393F0A9.1F758C5B (05:50:33.122 UTC Mon Dec 28 2020)
clock offset is 1343.7280 msec, root delay is 49.13 msec
root dispersion is 2275.31 msec, peer dispersion is 3.44 msec
loopfilter state is 'SPIK' (Spike), drift is 0.000499999 s/s
system poll interval is 64, last update was 173 sec ago.
```

stratum 2 because it’s synchronizing from Google (stratum 1)
(To show NTP clock details)

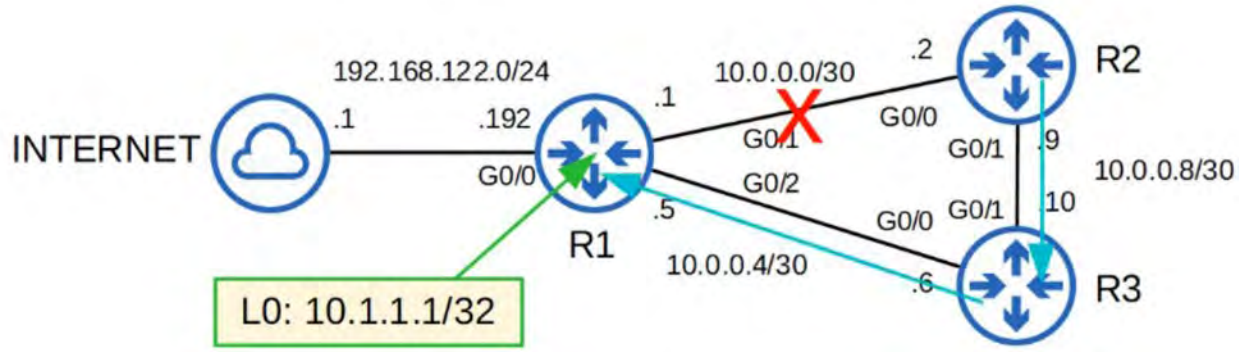
```
R1(config)#do show clock detail
06:56:32.315 UTC Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
05:23:06 UTC Mon Dec 28 2020
R1(config)#clock timezone JST 9
R1(config)#ntp update-calendar
R1(config)#do show clock detail
15:57:33.078 JST Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
15:57:36 JST Mon Dec 28 2020
```

This command configures the ROUTER to update the HARDWARE CLOCK (Calendar) with the time learned via NTP
R1(config)# ntp update-calendar
The HARDWARE CLOCK tracks the DATE and TIME on the DEVICE - even if it restarts, power is lost, etc.
When the SYSTEM is restarted, the HARDWARE CLOCK is used to INITIALIZE the SOFTWARE CLOCK

CONFIGURE A LOOPBACK INTERFACE FOR AN NTP SERVER

```
R1(config)#interface loopback0
R1(config-if)#ip address 10.1.1.1 255.255.255.255
R1(config-if)#exit
R1(config)#ntp source loopback0
```

Why configure a LOOPBACK DEVICE on R1 for NTP ?
If one of R1's ROUTER INTERFACES goes down, it will still be accessible via R3's ROUTING path



SET NTP SERVER for R2 using the LOOPBACK INTERFACE on R1

```
R2(config)#ntp server 10.1.1.1
R2(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.1.1.1        216.239.35.12    2      0     64     1  7.038 -13.128 3937.5
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R2(config)#do show ntp status
Clock is synchronized, stratum 3, reference is 10.1.1.1
...
```

SETTING R3 NTP SOURCE SERVERS using R1 and R2

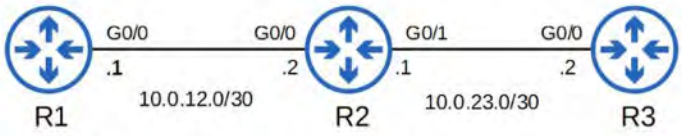
```
R3(config)#ntp server 10.1.1.1
R3(config)#ntp server 10.2.2.2
R3(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset    disp
*~10.1.1.1        216.239.35.0     2      1     64     0  0.000   0.000 15937.
 ~10.2.2.2        10.1.1.1         3      1     64     0  0.000   0.000 15937.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

Servers with lower stratum levels are preferred.

NOTE : R1 has PREFERENCE because it's STRATUM TIER is HIGHER than R2s

---

CONFIGURING NTP SERVER MODE




```
R1(config)#ntp ?
  access-group        Control NTP access
  allow               Allow processing of packets
  authenticate        Authenticate time sources
  authentication-key  Authentication key for trusted time sources
  broadcastdelay      Estimated round-trip delay
  clock-period        Length of hardware clock tick
  logging             Enable NTP message logging
  master              Act as NTP master clock
  max-associations    Set maximum number of associations
  maxdistance         Maximum Distance for synchronization
  mindistance         Minimum distance to consider for clockhop
  orphan              Threshold Stratum for orphan mode
  panic               Reject time updates > panic threshold (default 1000Sec)
  passive             NTP passive mode
  peer                Configure NTP peer
  server              Configure NTP server
  source              Configure interface for source address
  trusted-key         Key numbers for trusted time sources
  update-calendar     Periodically update calendar with NTP time
```

```
R1(config)#ntp master ?
  <1-15>  Stratum number
  <cr>

R1(config)#ntp master
R1(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset    disp
*~127.127.1.1     .LOCL.           7      2     16   377  0.000   0.000   0.292
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R1(config)#do show ntp status
Clock is synchronized, stratum 8, reference is 127.127.1.1
...
```

The default stratum of the **ntp master** command is 8.

```
R2(config)#ntp server 10.0.12.1

R2(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.0.12.1       127.127.1.1      8      2     64     1  5.263  62.494 187.64
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
R3(config)#ntp server 10.0.12.1

R3(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.0.12.1       127.127.1.1      8     45     64    17 21.534 -21.440  0.976
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

CONFIGURING NTP SYMMETRIC ACTIVE MODE

Command to configure NTP SYMMETRIC MODE R2(config)#ntp peer <peer ip address>

```
R2(config)#ntp peer 10.0.23.2

R2(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.0.12.1       127.127.1.1      8     60     64    17 24.040 206.682  0.987
 ~10.0.23.2       10.0.12.1        9     33     64     0  0.000   0.000 15937.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
R3(config)#ntp peer 10.0.23.1

R3(config)#do show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.0.12.1       127.127.1.1      8     11     64    37 12.605  -7.406 63.575
 ~10.0.23.1       10.0.12.1        9      1     64     0  0.000   0.000 15937.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

CONFIGURE NTP AUTHENTICATION

- NTP AUTHENTICATION can be configured, although it is OPTIONAL
- It allows NTP CLIENTS to ensure they only sync to the intended SERVERS
- To CONFIGURE NTP AUTHENTICATION:
  - ntp authenticate (Enables NTP AUTHENTICATION)
  - ntp authenticate-key *key-number* md5 *key* (Create the NTP AUTHENTICATION Key(s))
  - ntp trusted-key *key-number* (Specify the Trusted Key(s))
  - ntp server *ip-address* key *key-number* (Specify which key to use for the server)

EXAMPLE CONFIGURATIONS

```
R1(config)#ntp authenticate
R1(config)#ntp authentication-key 1 md5 jeremysitlab
R1(config)#ntp trusted-key 1
```

```
R2(config)#ntp authenticate
R2(config)#ntp authentication-key 1 md5 jeremysitlab
R2(config)#ntp trusted-key 1
R2(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.2 key 1
```

```
R3(config)#ntp authenticate
R3(config)#ntp authentication-key 1 md5 jeremysitlab
R3(config)#ntp trusted-key 1
R3(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.1 key 1
```

NTP COMMAND REVIEW

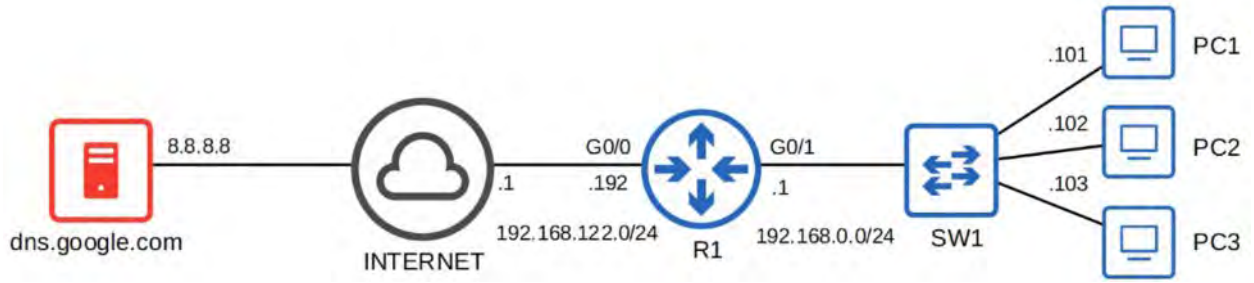
```
!Basic Configuration Commands

R1(config)# ntp server ip-address [prefer]
R1(config)# ntp peer ip-address
R1(config)# ntp update-calendar
R1(config)# ntp master [stratum]
R1(config)# ntp source interface

!Basic Show Commands

R1# show ntp associations
R1# show ntp status

!Basic Authentication Commands

R1(config)# ntp authenticate
R1(config)# ntp authentication-key key-number md5 key
R1(config)# ntp trusted-key key-number
R1(config)# ntp server ip-address key key-number
R1(config)# ntp peer ip-address key key-number
```

38. DNS (Domain Name System)

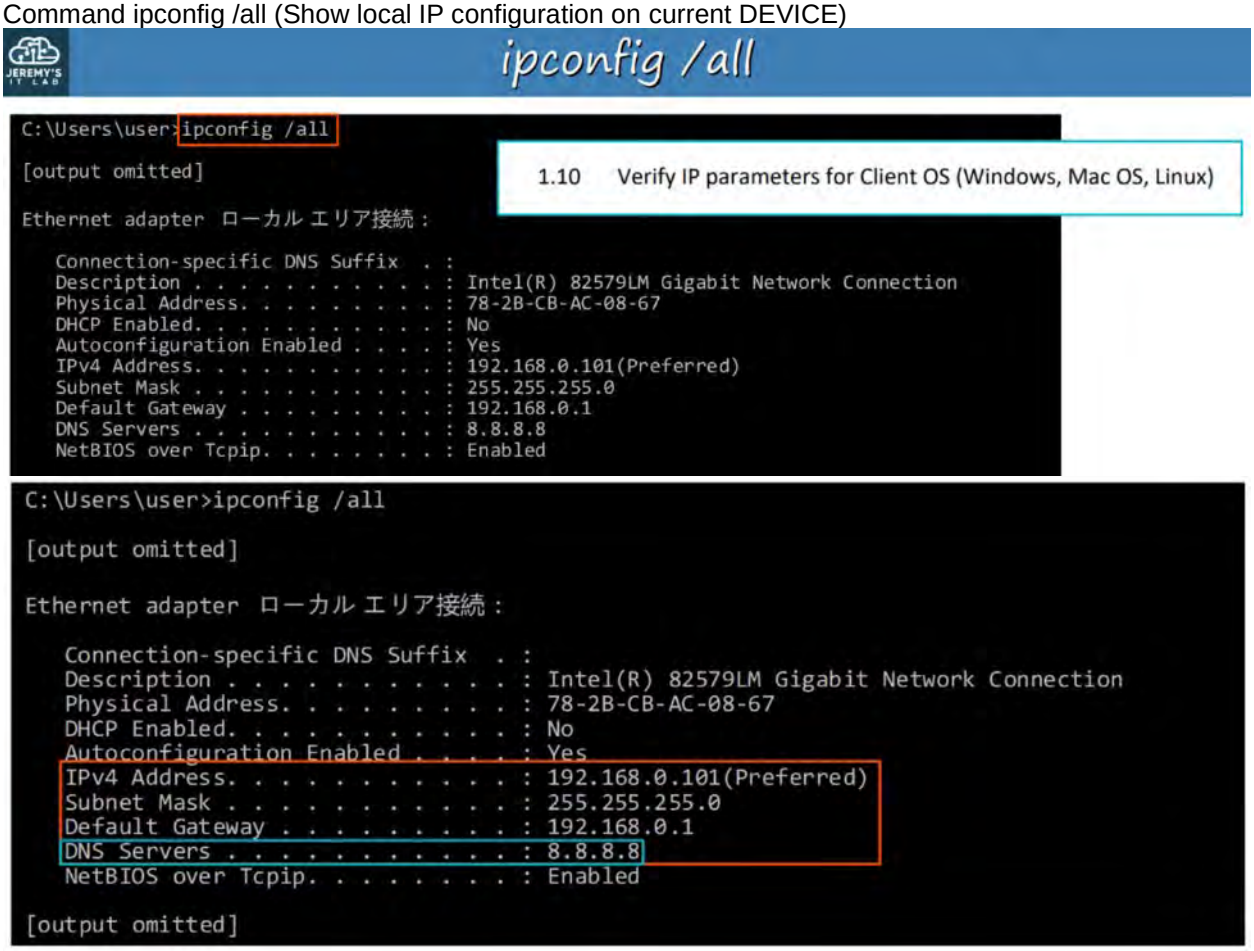
THE PURPOSE OF DNS

- 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 is 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

BASIC FUNCTIONS OF DNS



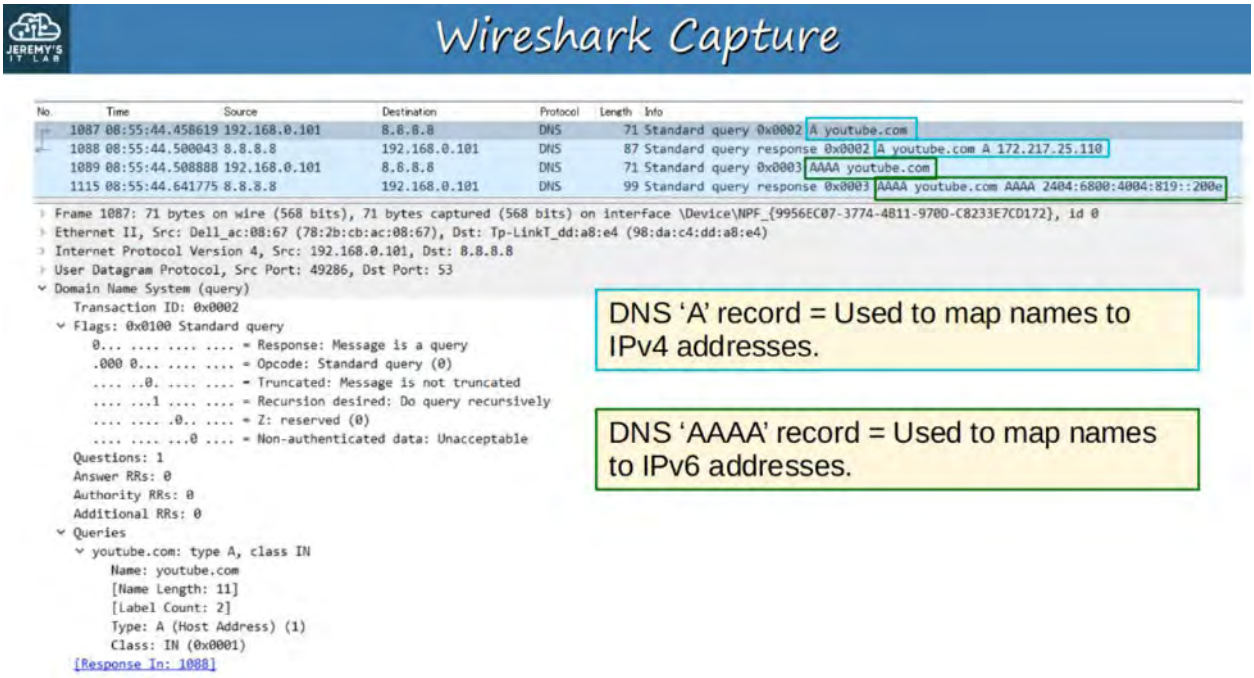
Command ipconfig /all (Show local IP configuration on current DEVICE)

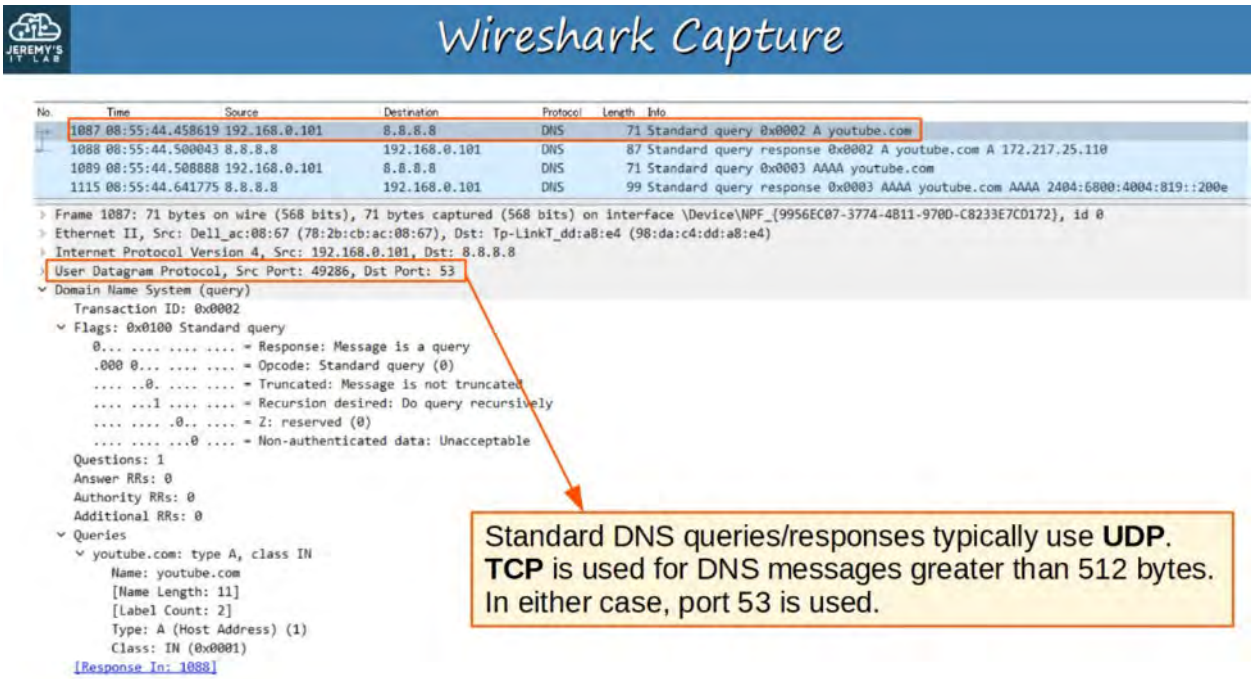


Command nslookup (Shows IP information for a given DNS entry)

```
C:\Users\user>nslookup youtube.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    youtube.com
Addresses:  2404:6800:4004:819::200e
          172.217.25.110

C:\Users\user>ping youtube.com

Pinging youtube.com [172.217.25.110] with 32 bytes of data:
Reply from 172.217.25.110: bytes=32 time=10ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117

Ping statistics for 172.217.25.110:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 10ms, Average = 7ms
```

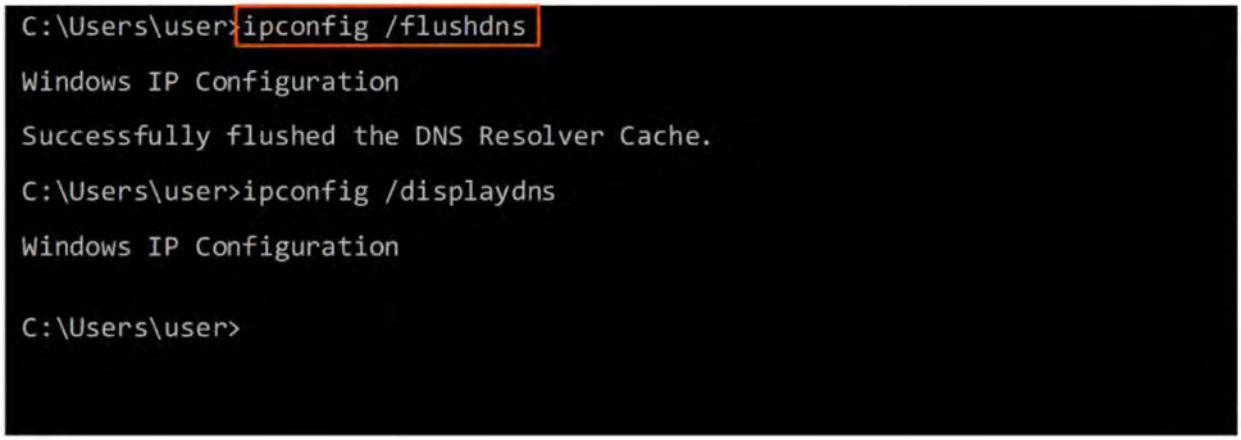
WIRESHARK CAPTURE of above COMMANDS



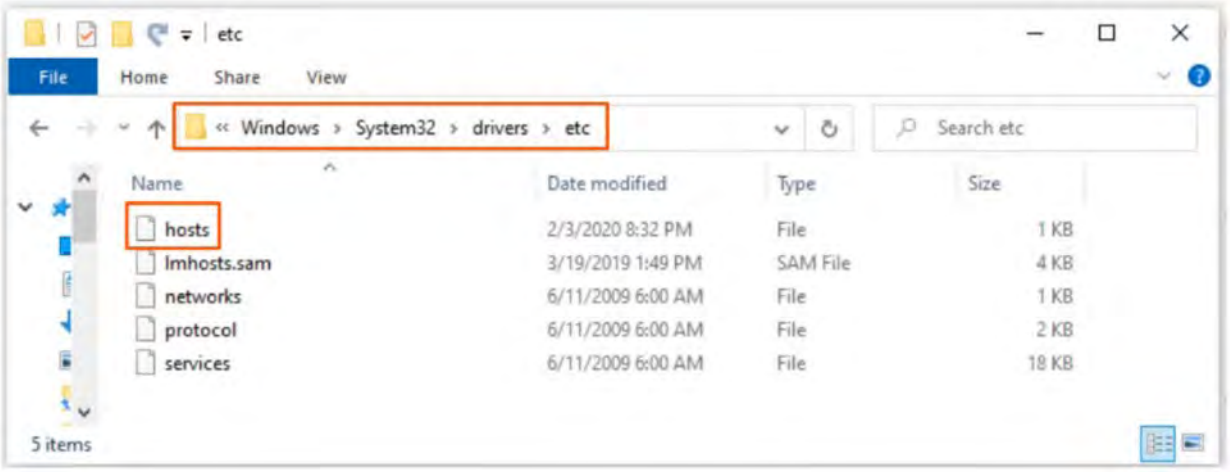


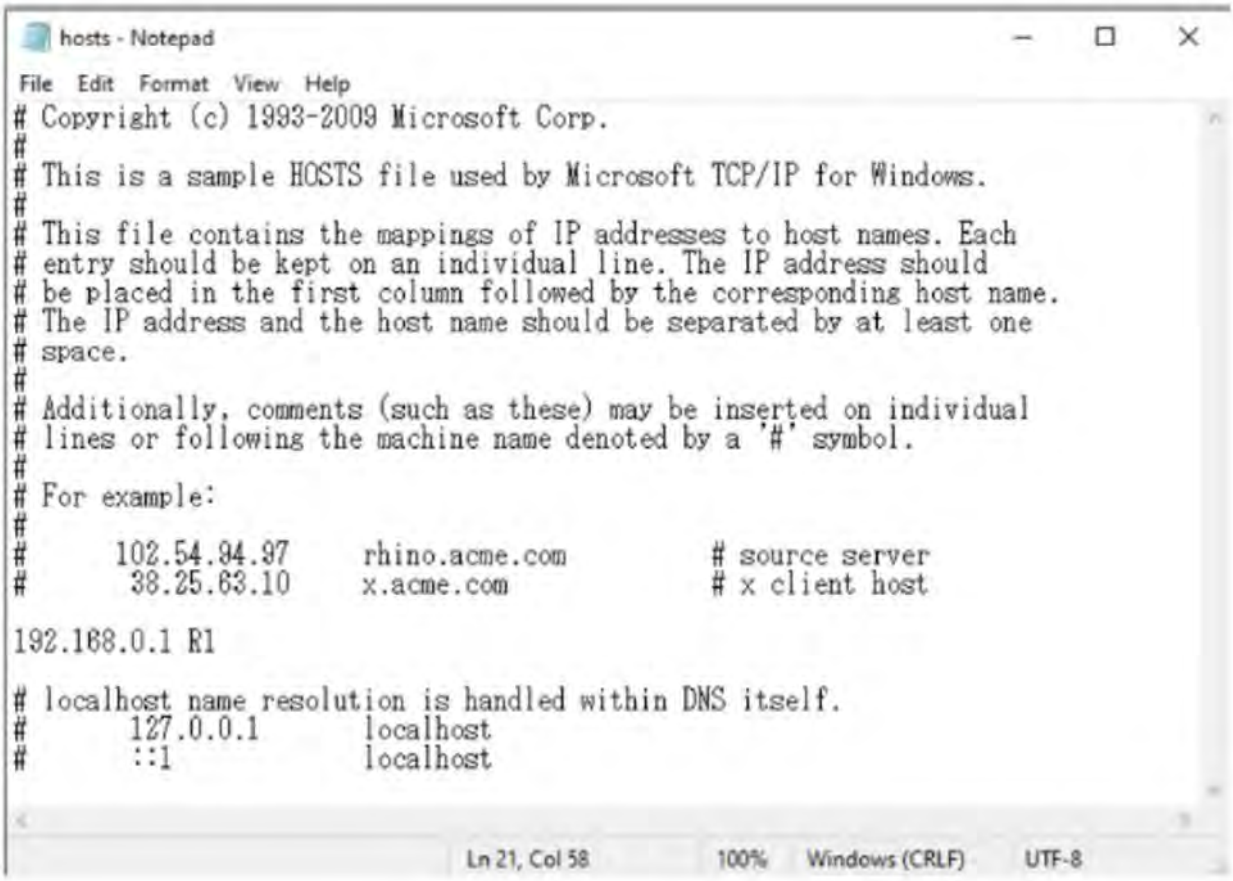
Command ipconfig /displaydns (Displays DNS cache)

Command ipconfig /flushdns (Clears DNS cache)



HOSTS Files
WINDOWS HOSTS location





CONFIGURING DNS IN CISCO IOS

- For HOSTS in a NETWORK to use DNS, you don't need to configure DNS on the ROUTERS.
  - They will simply FORWARD the DNS messages like any other packets
- However, a CISCO ROUTER can be configured as a DNS SERVER, although it's rare
  - If an INTERNAL DNS SERVER is used, usually it's a WINDOWS or LINUX SERVER
- A CISCO ROUTER can also be configured as a DNS CLIENT

Command ip dns server and ip host <hostname> <ip address>

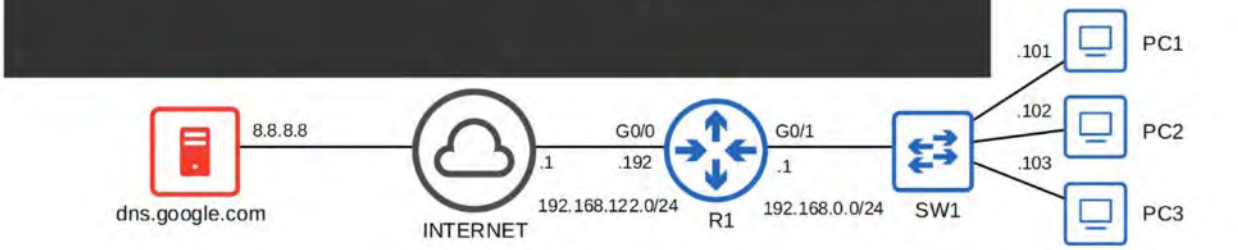
```
R1(config)#ip dns server

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.

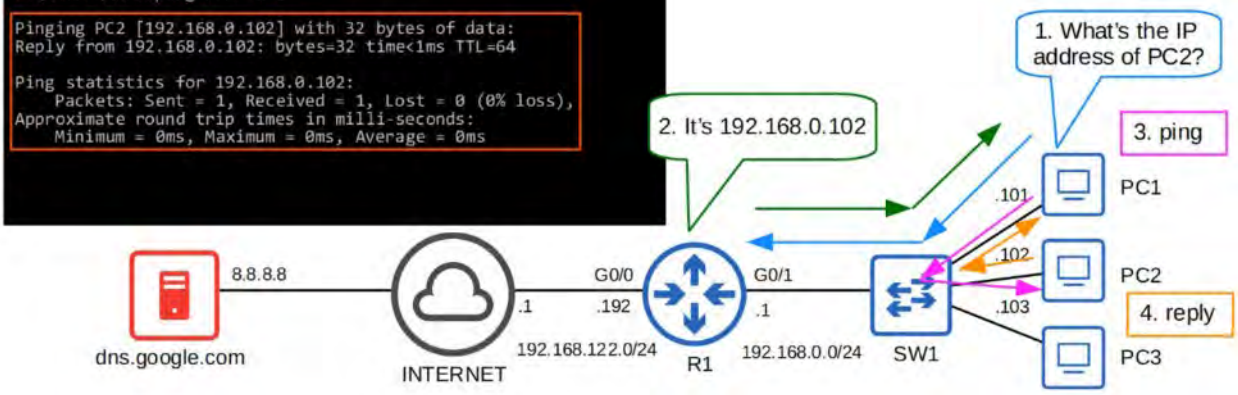
Configure a list of hostname/IP address mappings.

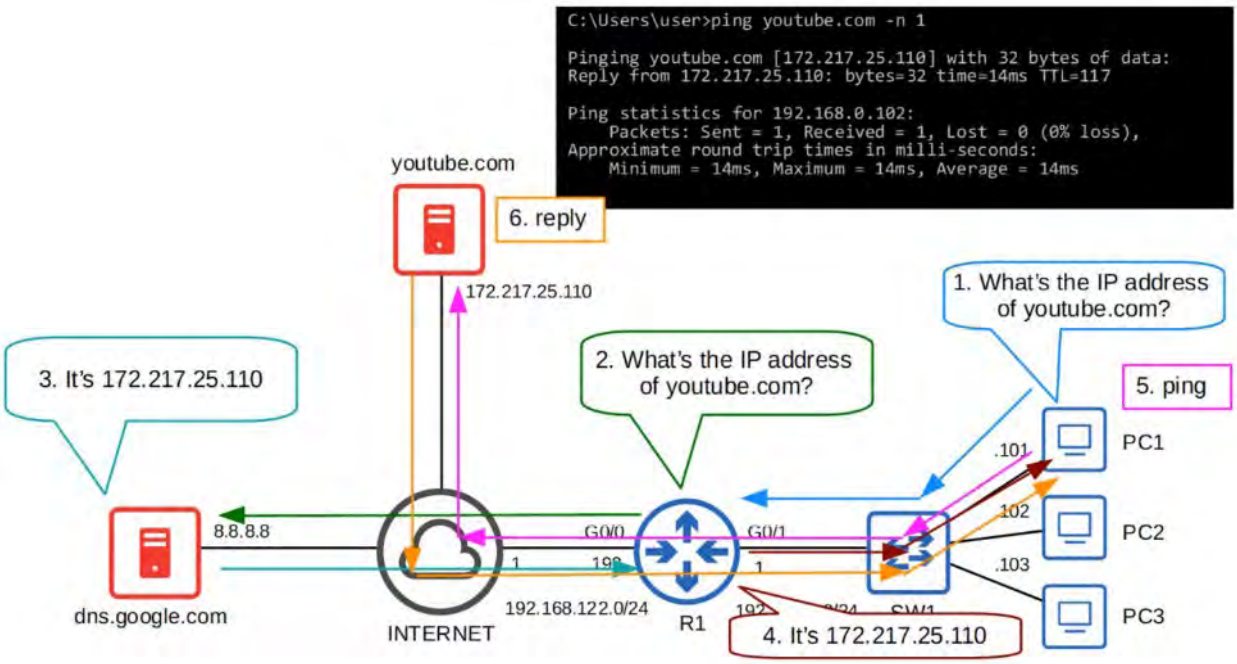
Configure a DNS server that R1 will query if the requested record isn't in its host table.

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




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

[output omitted]

   IPv4 Address. . . . . . . . . . . : 192.168.0.101(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]

C:\Users\user>ping PC2 -n 1

Pinging PC2 [192.168.0.102] with 32 bytes of data:
Reply from 192.168.0.102: bytes=32 time<1ms TTL=64

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

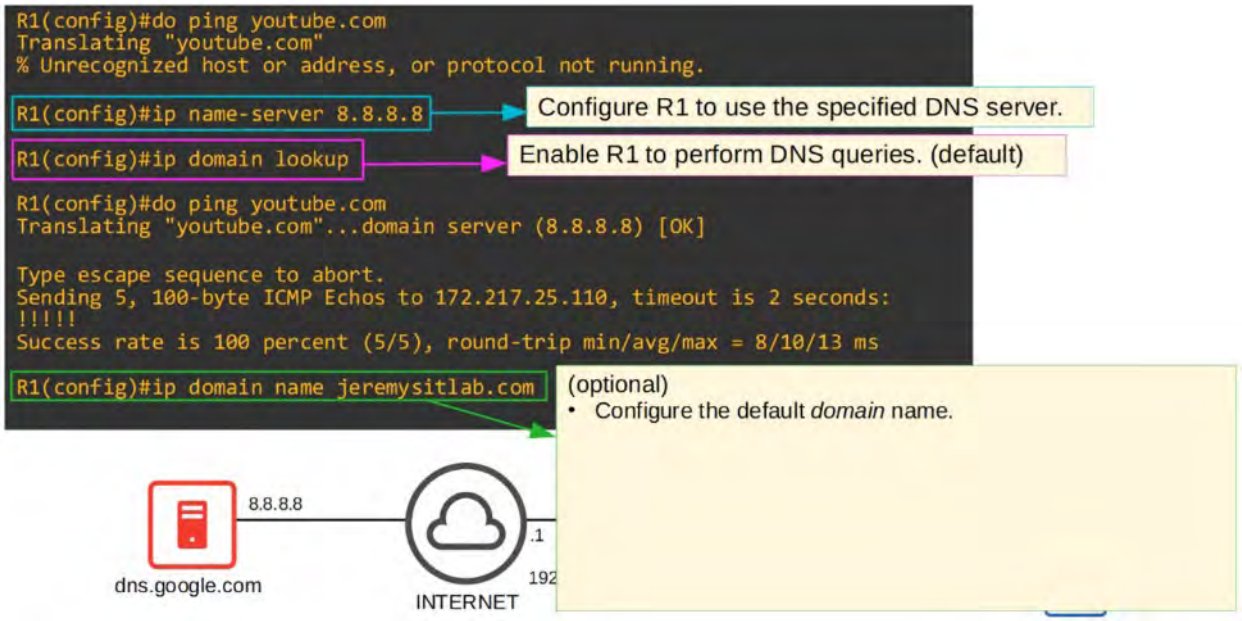




Command show hosts

```
R1#show hosts
Default domain is not set
Name/address lookup uses domain service
Name servers are 8.8.8.8

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
youtube.com               None  (temp, OK)  0   IP     172.217.25.110
R1                        None  (perm, OK)  4   IP     192.168.0.1
PC1                       None  (perm, OK)  1   IP     192.168.0.101
PC2                       None  (perm, OK)  4   IP     192.168.0.102
PC3                       None  (perm, OK)  4   IP     192.168.0.103
```

Command ip name-server and ip domain lookup



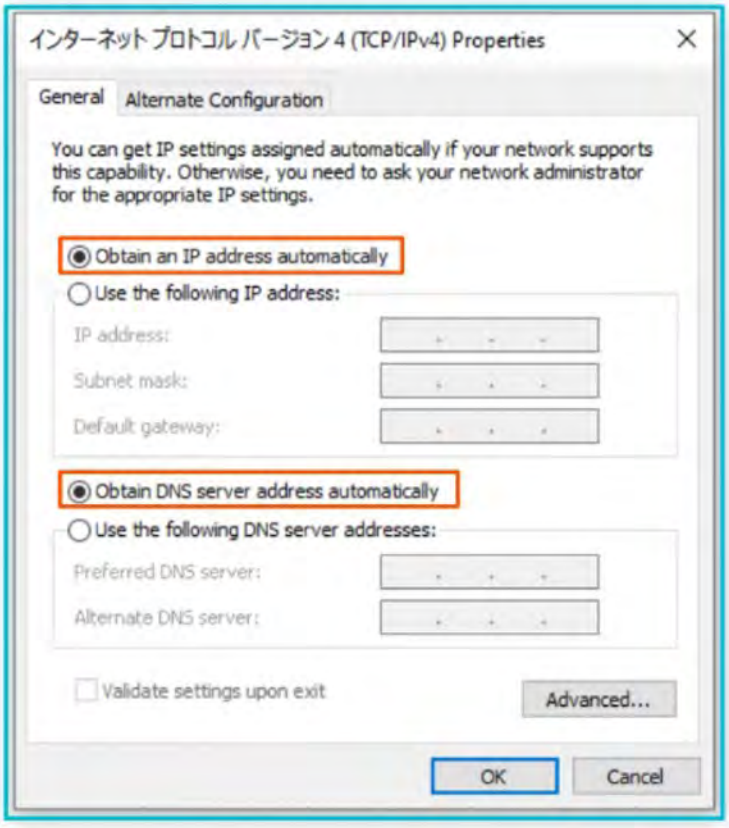
COMMAND REVIEW:

39. DHCP (Dynamic Host Configuration Protocol)

THE PURPOSE OF DHCP

- DHCP is used to allow HOSTS to automatically / dynamically learn various aspects of their NETWORK configuration; without MANUAL / STATIC configuration
- It is an ESSENTIAL part of modern NETWORKS
    - When you connect a phone / laptop to WiFi, do you ask your NETWORK admin which IP ADDRESS, SUBNET MASK, DEFAULT GATEWAY, etc the phone / laptop should use ?
- Typically used for CLIENT devices (workstations, 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 LARGE NETWORKS, the DHCP SERVER is usually a Windows / Linux SERVER

BASIC FUNCTIONS OF DHCP



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]
```

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.

The Basic Functions of DHCP

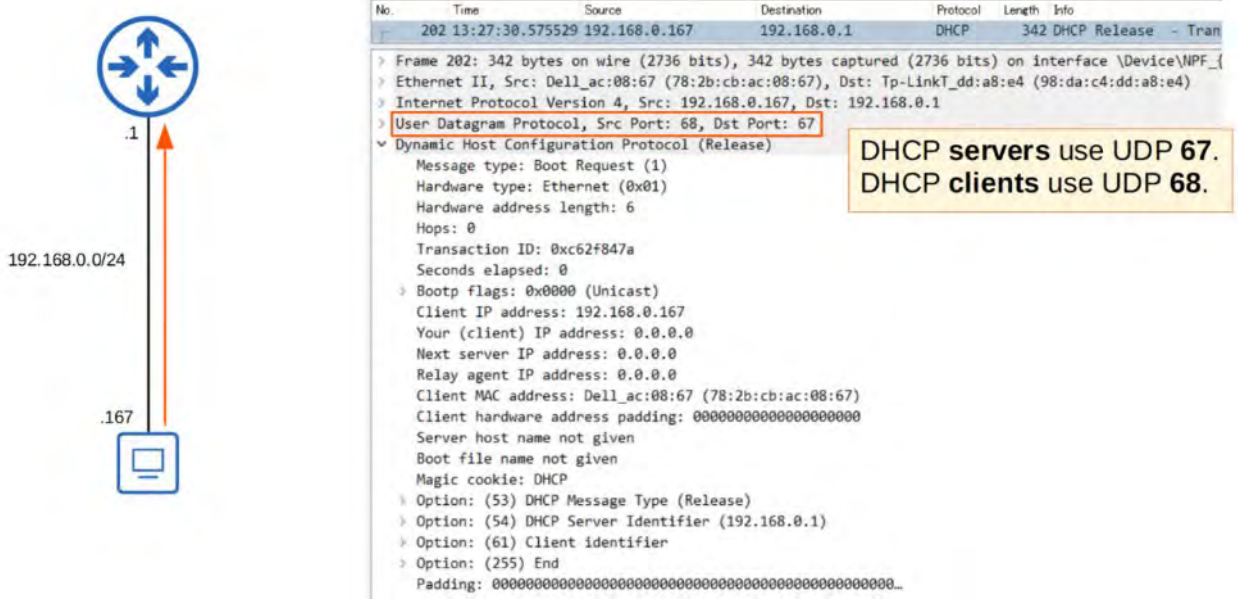
```
C:\Users\user>ipconfig /all

[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]
```

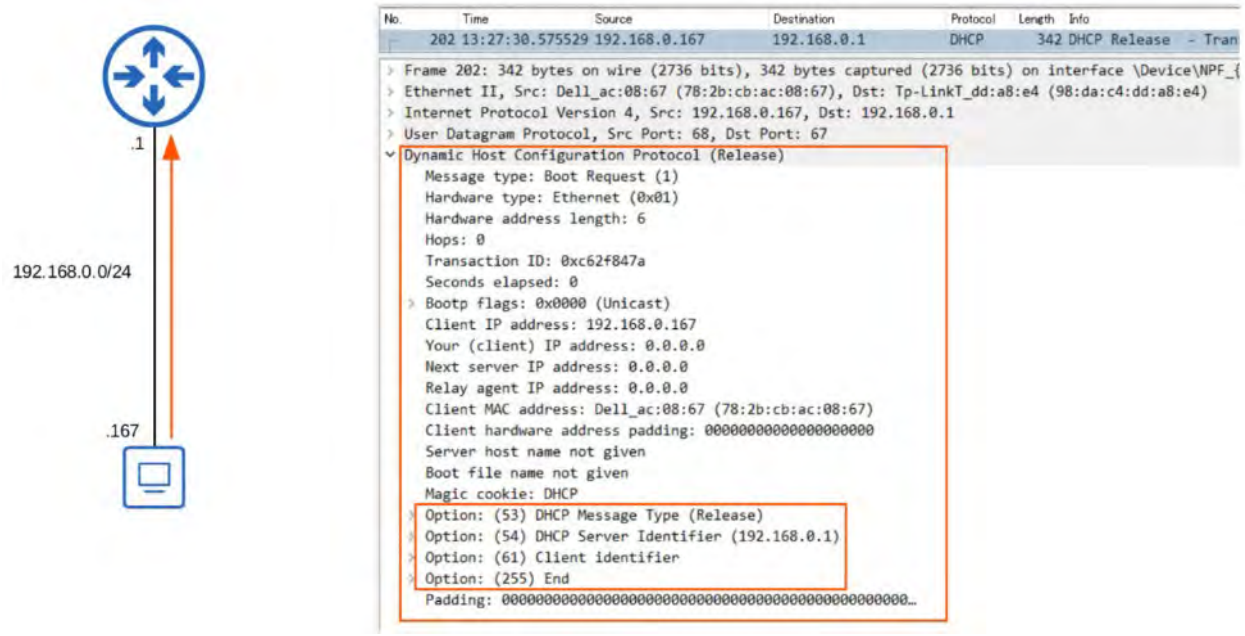
Note: ALL the IPs are the same because this is Jeremy's Home ROUTER (it provides all these services)
Command ipconfig /release

ipconfig /release

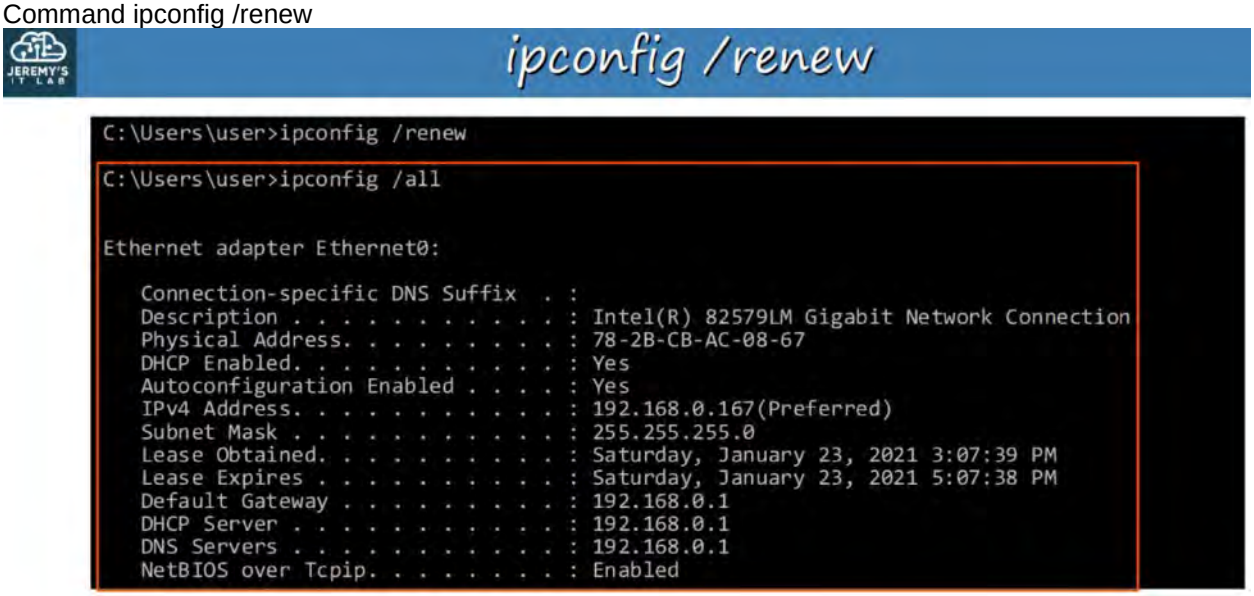
```
C:\Users\user>ipconfig /release

Windows IP Configuration

[output omitted]

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Default Gateway . . . . . . . . . :

[output omitted]
```

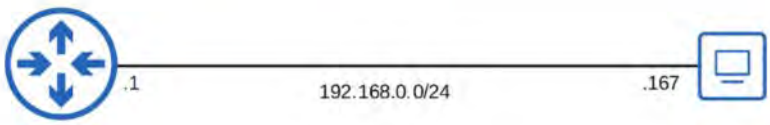
Wireshark capture of the ipconfig /release mechanism



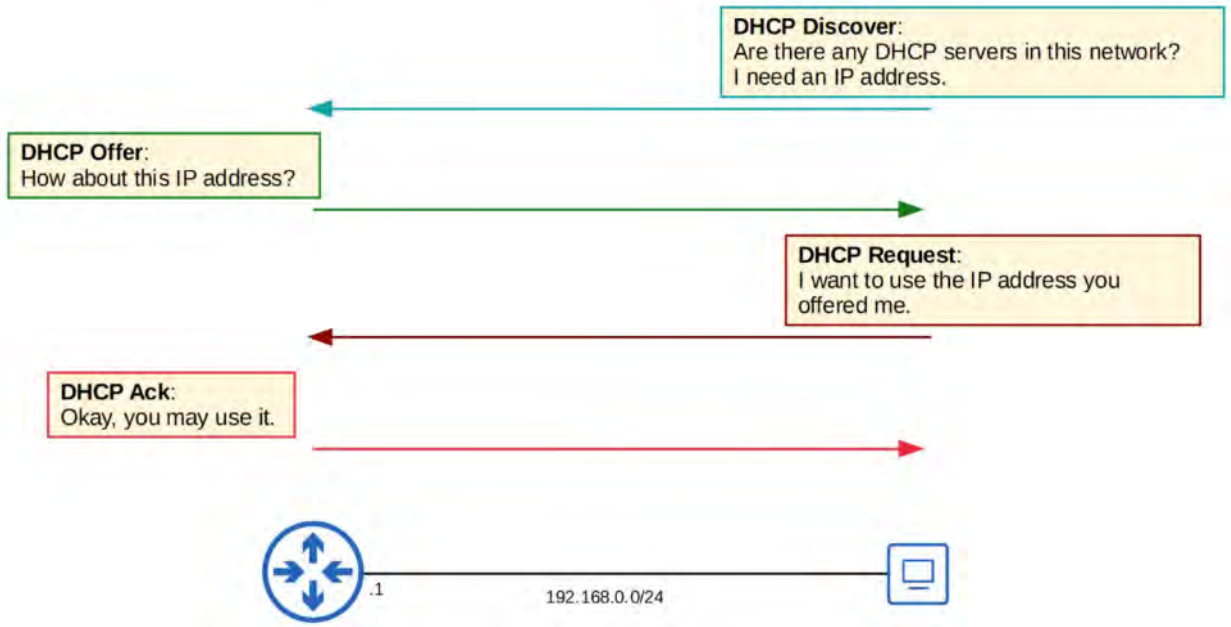


Command ipconfig /renew



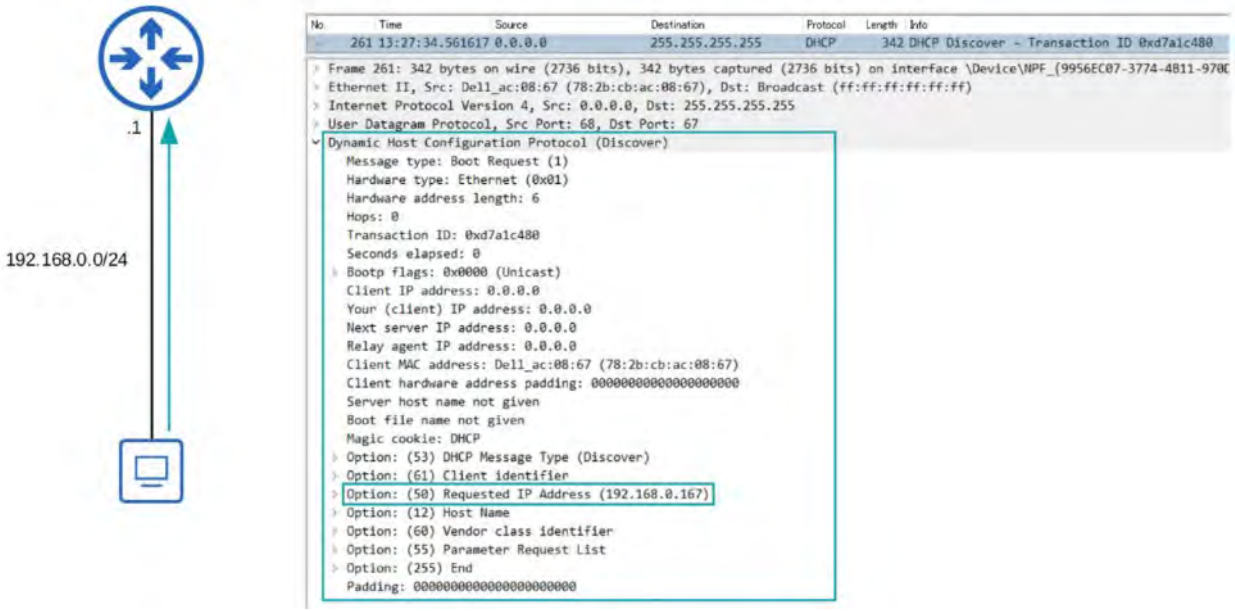


Renewing Process has FOUR messages:



1. DHCP DISCOVER

- Are there any DHCP Servers in this NETWORK? I need an IP ADDRESS ?



NOTE the use of DHCP Reserved Ports 67 and 68

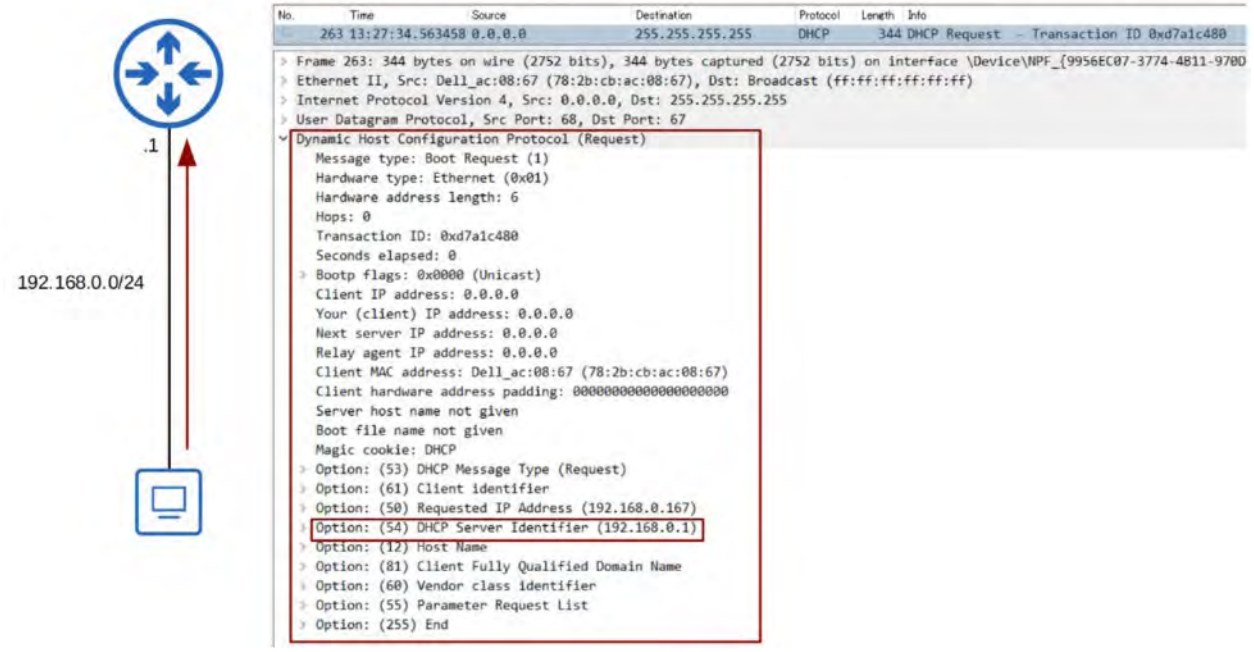
2. DHCP OFFER:

- How about THIS IP ADDRESS ?

- The DHCP OFFER message can be either BROADCAST or UNICAST
- NOTE OPTIONS at the bottom : Message Type, Server ID, Lease Time, Subnet, etc.

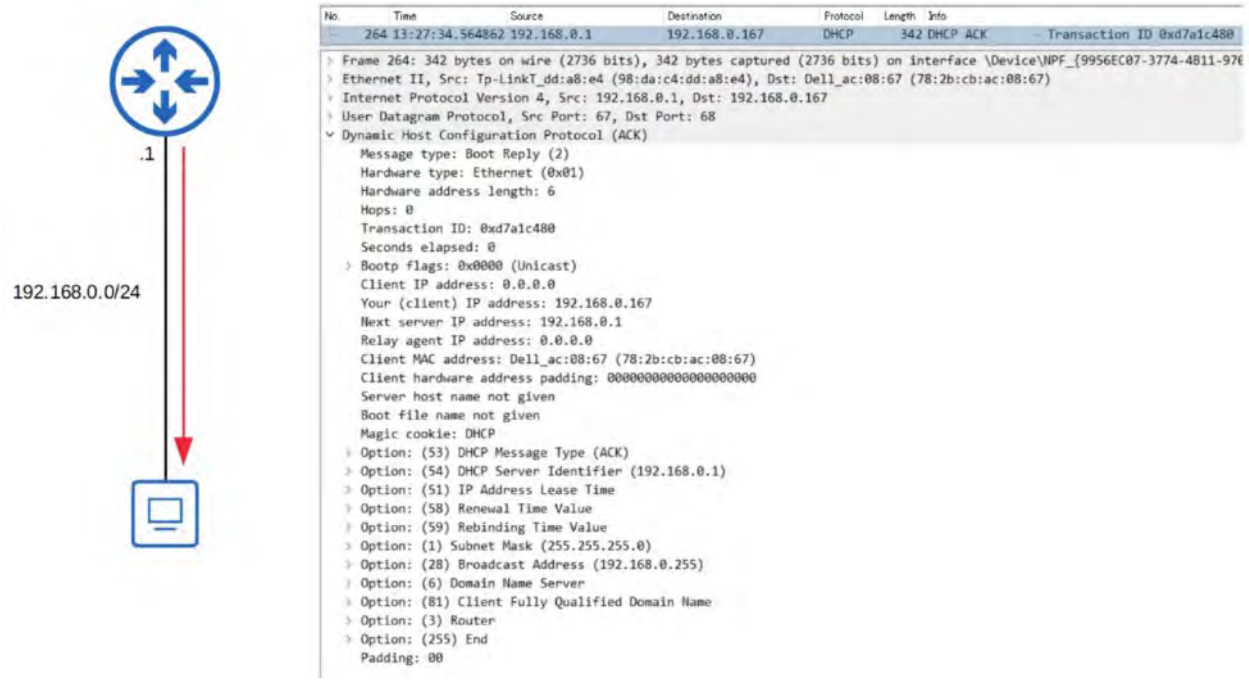
3. DHCP REQUEST

- I want to use the IP ADDRESS that was offered

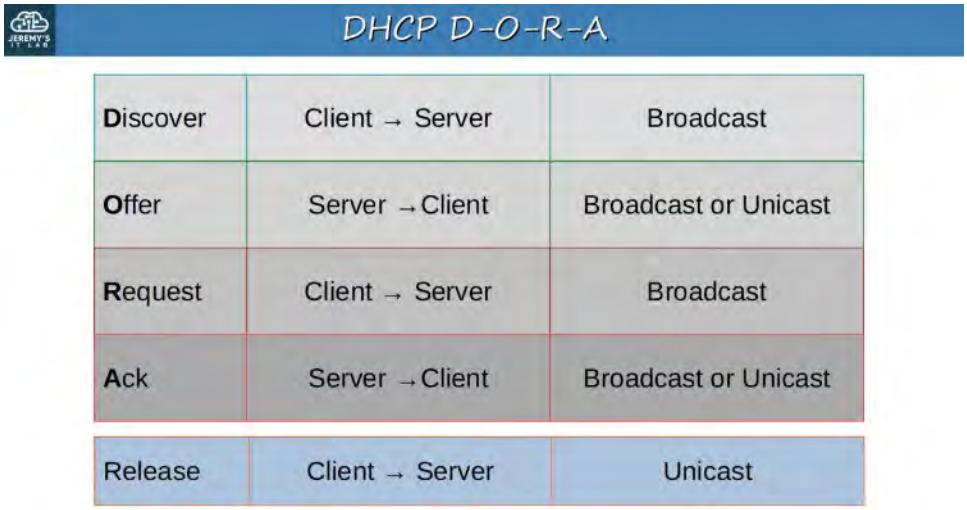


4. DHCP ACK

- Okay! You may use THAT ADDRESS

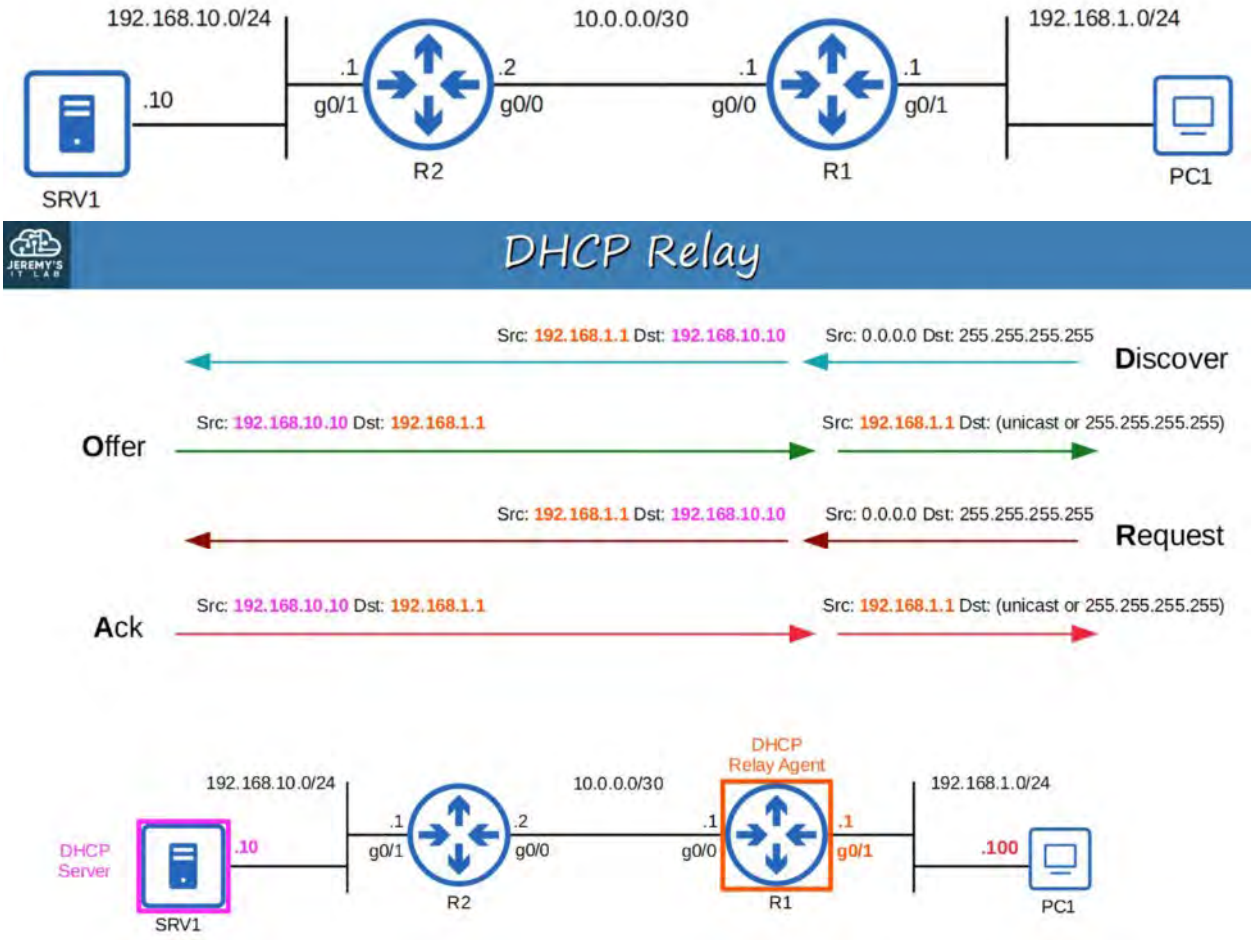


DHCP RENEW PROCESS SUMMARY

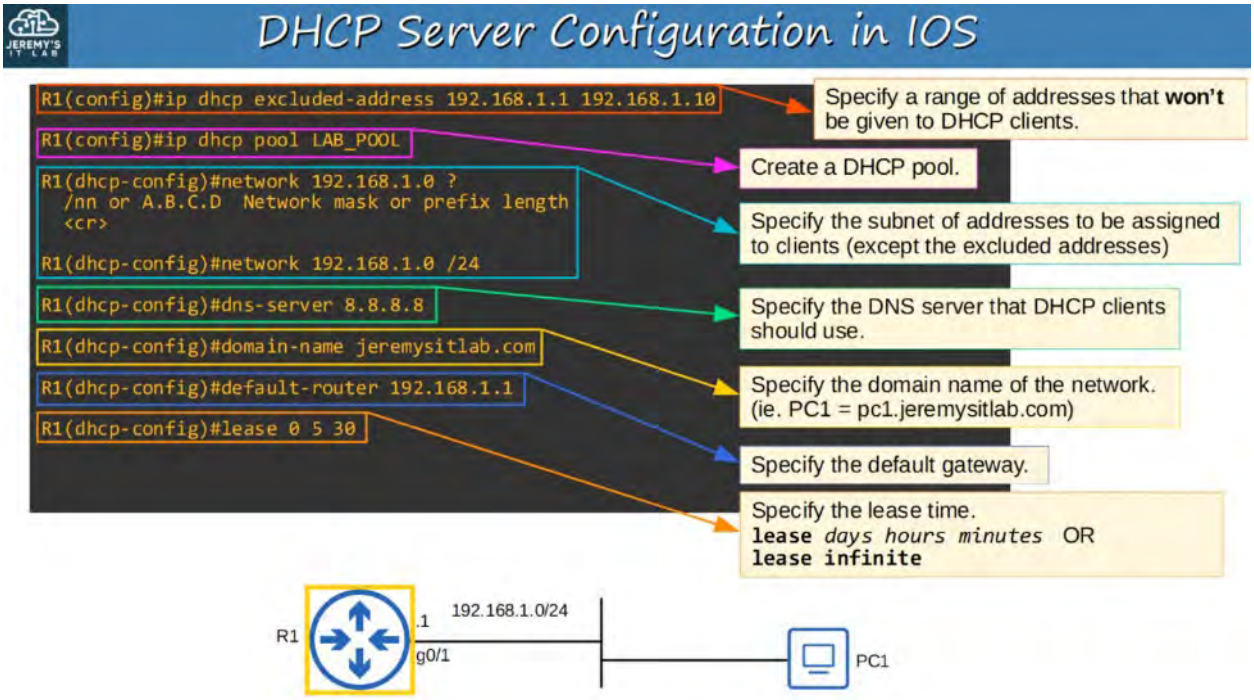


DHCP RELAY

- Some NETWORK engineers might choose to configure each ROUTER to act as the DHCP SERVER for its connected LANS
- However, large enterprises often choose to use a CENTRALIZED DHCP SERVER
- If the SERVER is centralized, it won't receive the DHCP CLIENTS' Broadcast DHCP messages
- To FIX this, you can configure a ROUTER to act as a DHCP RELAY AGENT
- The ROUTER will forward the clients' Broadcast DHCP messages to the remote DHCP SERVER as a Unicast messages



CONFIGURING DHCP IN CISCO IOS
Commands for configuring DHCP SERVERS in Cisco IOS

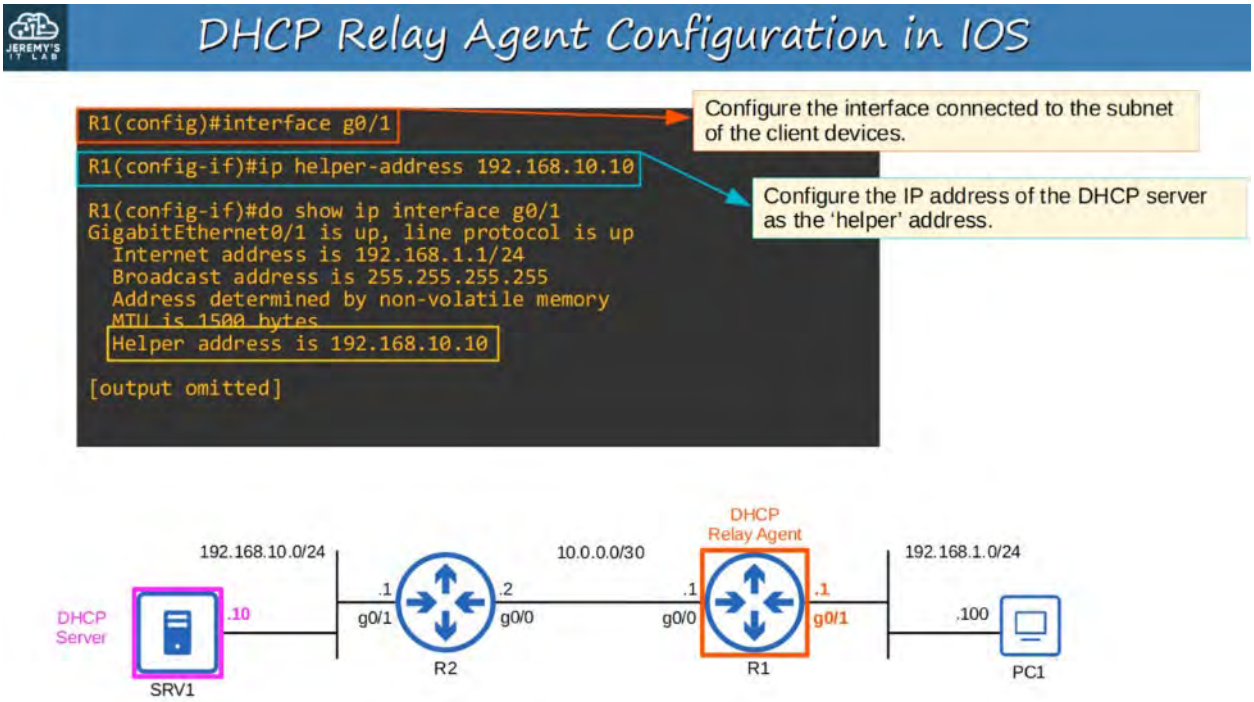


Command show ip dhcp binding

```
R1#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
192.168.1.11        0100.0c29.e727.39       Jan 24 2021 10:52 AM    Automatic
```

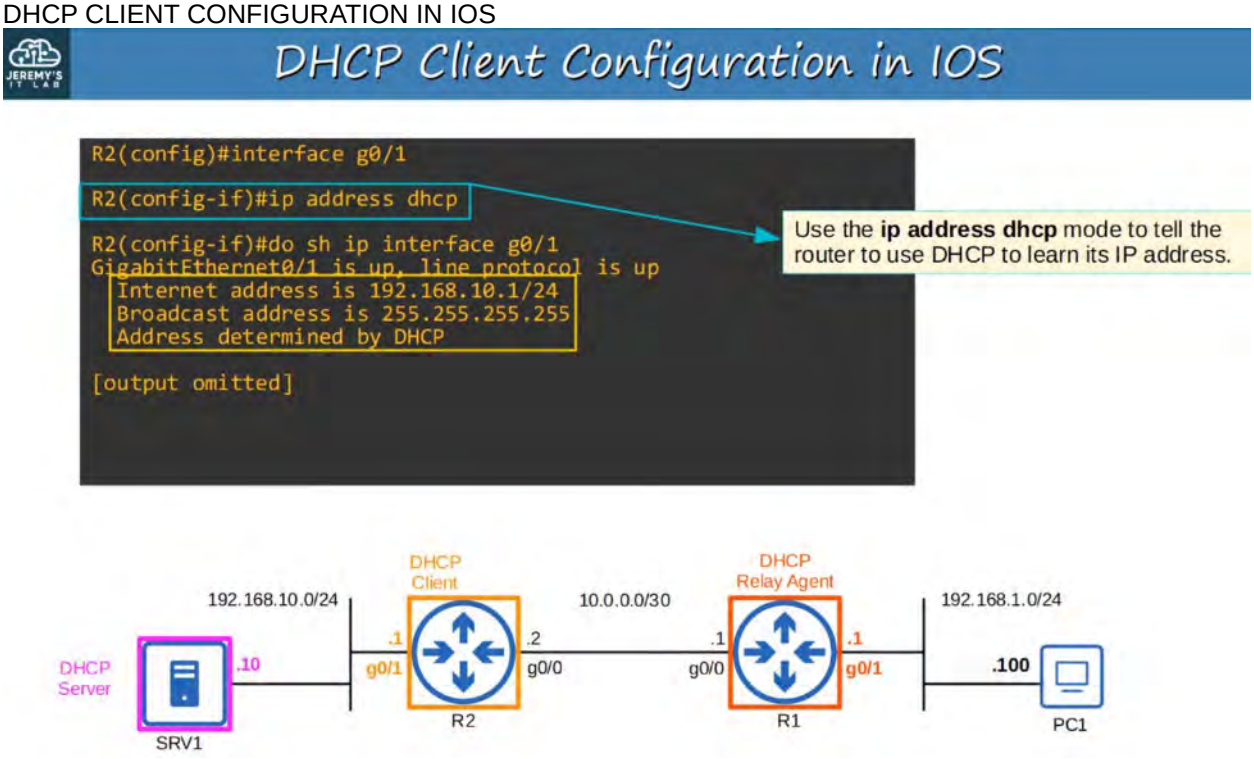
```
C:\Users\user>ipconfig /all

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : jeremysitlab.com
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection #2
   Physical Address. . . . . . . . . : 00-0C-29-E7-27-39
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.1.11(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Saturday, January 24, 2021 2:22:35 PM
   Lease Expires . . . . . . . . . . : Saturday, January 24, 2021 7:52:35 PM
   Default Gateway . . . . . . . . . : 192.168.1.1
   DHCP Server . . . . . . . . . . . : 192.168.1.1
   DNS Servers . . . . . . . . . . . : 8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

DHCP RELAY AGENT CONFIGURATION IN IOS



RELAY AGENT MUST HAVE CONNECTIVITY WITH DHCP SERVER

DHCP CLIENT CONFIGURATION IN IOS



COMMANDS SUMMARY

Command Summary

```
C:\Users\user> ipconfig /release
C:\Users\user> ipconfig /renew
```

```
R1(config)# ip dhcp excluded-address low-address high-address
R1(config)# ip dhcp pool pool-name
R1(dhcp-config)# network ip-address {/prefix-length | subnet-mask}
R1(dhcp-config)# dns-server ip-address
R1(dhcp-config)# domain-name domain-name
R1(dhcp-config)# default-router ip-address
R1(dhcp-config)# lease {days hours minutes | infinite}
R1# show ip dhcp binding
```

DHCP server

```
R1(config-if)# ip helper-address ip-address
```

DHCP relay agent

```
R1(config-if)# ip address dhcp
```

DHCP client

40. SNMP (Simple Network Management Protocol)

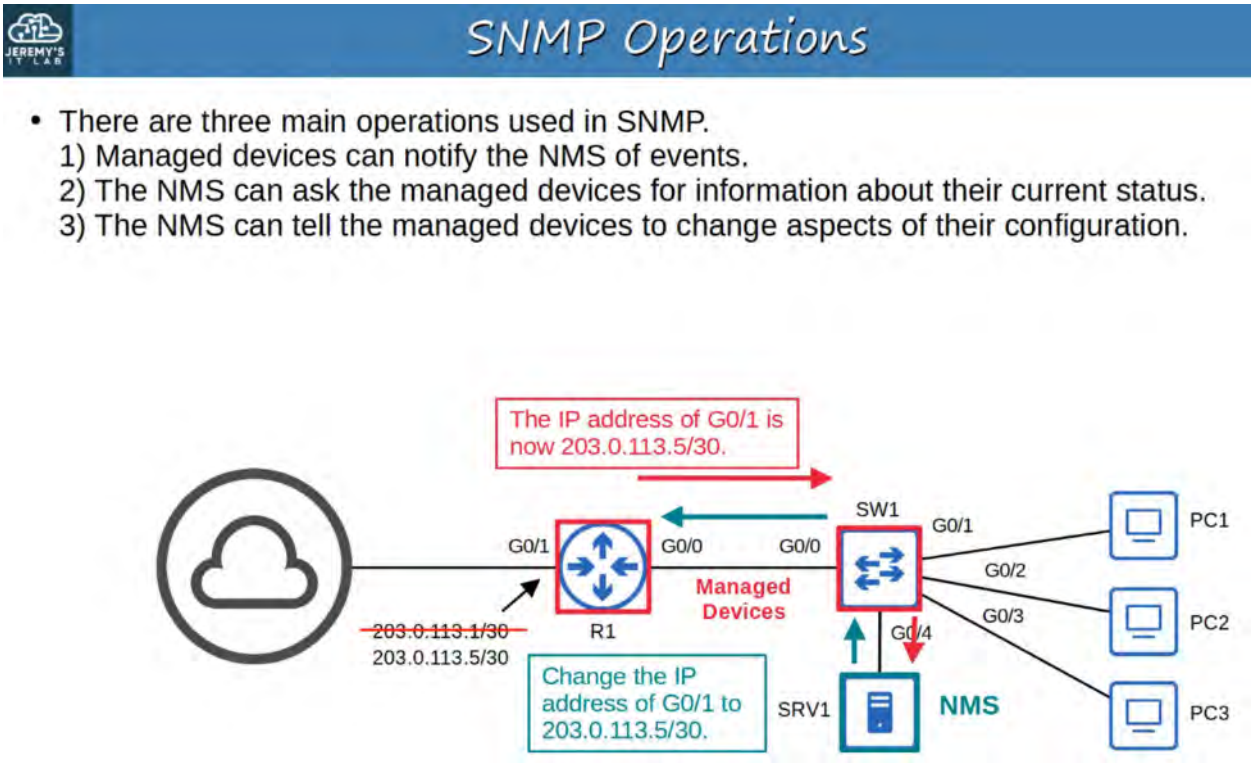
SNMP OVERVIEW

- SNMP is an INDUSTRY-STANDARD FRAMEWORK and PROTOCOL that was originally released in 1988

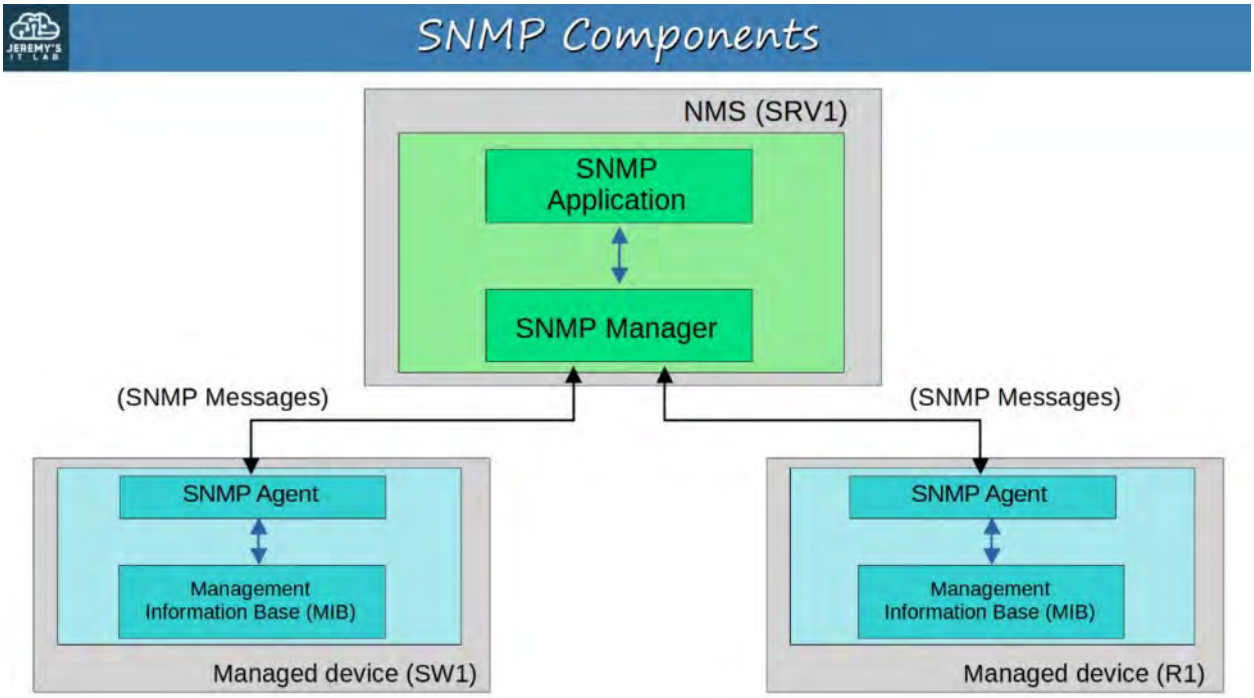
These RFCs make up SNMPv1 (Do not need to memorize)
RFC 1065 - Structure and identification of management information for TCP/IP based internets
RFC 1066 - Management information base for network management of TCP/IP based internets
RFC 1067 - A simple network management protocol

- 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:
    - MANAGED DEVICES
        - These are the DEVICES being managed using SNMP
            - Ex: ROUTERS, SWITCHES
    - NETWORK MANAGEMENT STATION (NMS)
        - The DEVICE / DEVICES managing the MANAGED DEVICES
        - THIS is the SNMP 'SERVER'

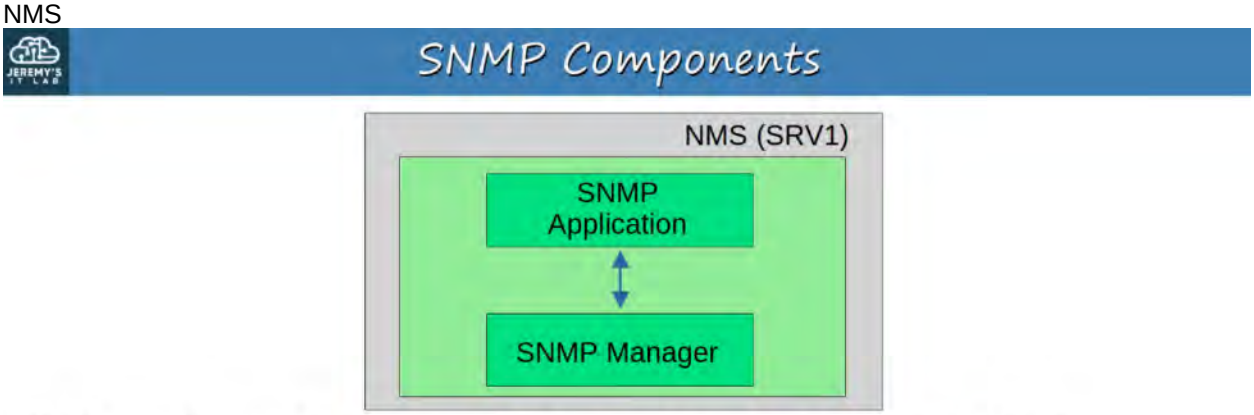
SMNP OPERATIONS



SMNP COMPONENTS OVERVIEW



NMS



- 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.

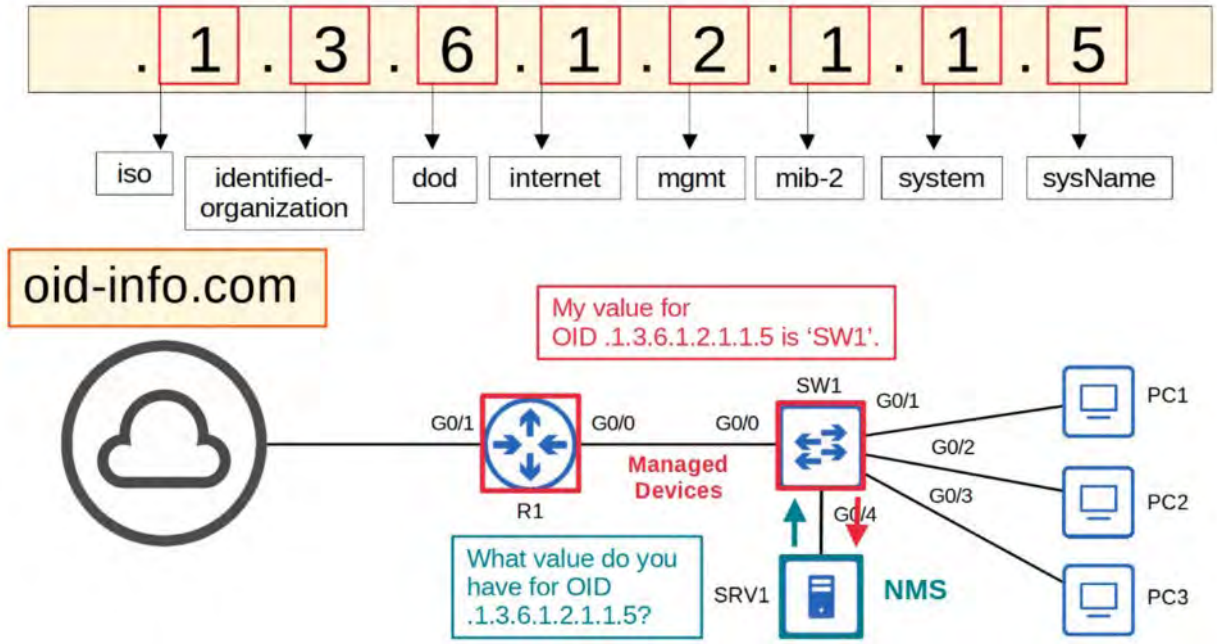
MANAGED DEVICES

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 SNMPv2

- **SNMPv3**
  - A much more SECURE version of SNMP that supports STRONG ENCRYPTION and AUTHENTICATION.

💡 WHENEVER POSSIBLE, this version should be used!

SNMP MESSAGES

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*

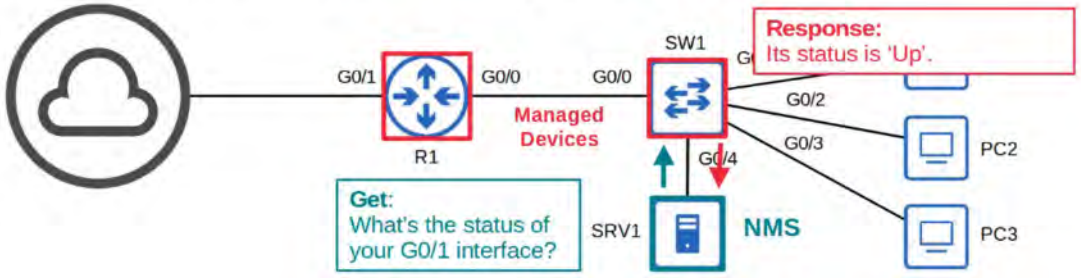
1. SNMP READ

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.

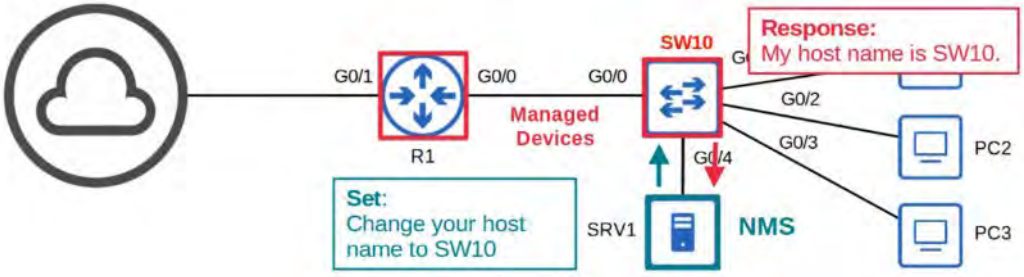
- **GetBulk**
  → A more efficient version of the **GetNext** message (introduced in SNMPv2).



2. SMNP WRITE

SNMP 'Write' Messages

- **Set**
  → A request sent from the manager to the agent to change the value of one or more variables. The agent will send a *Response* message with the new values.

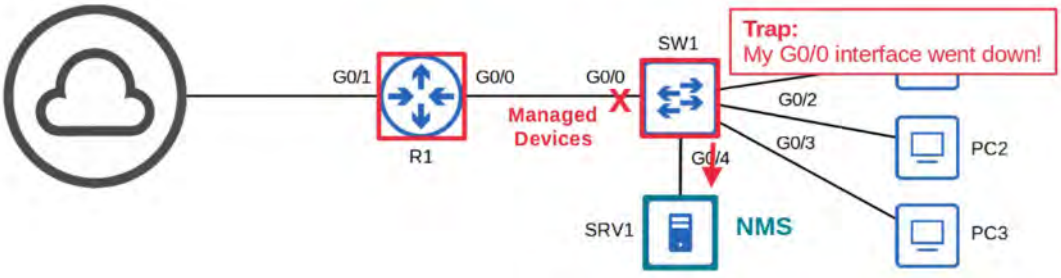


3. SNMP NOTIFICATION

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.



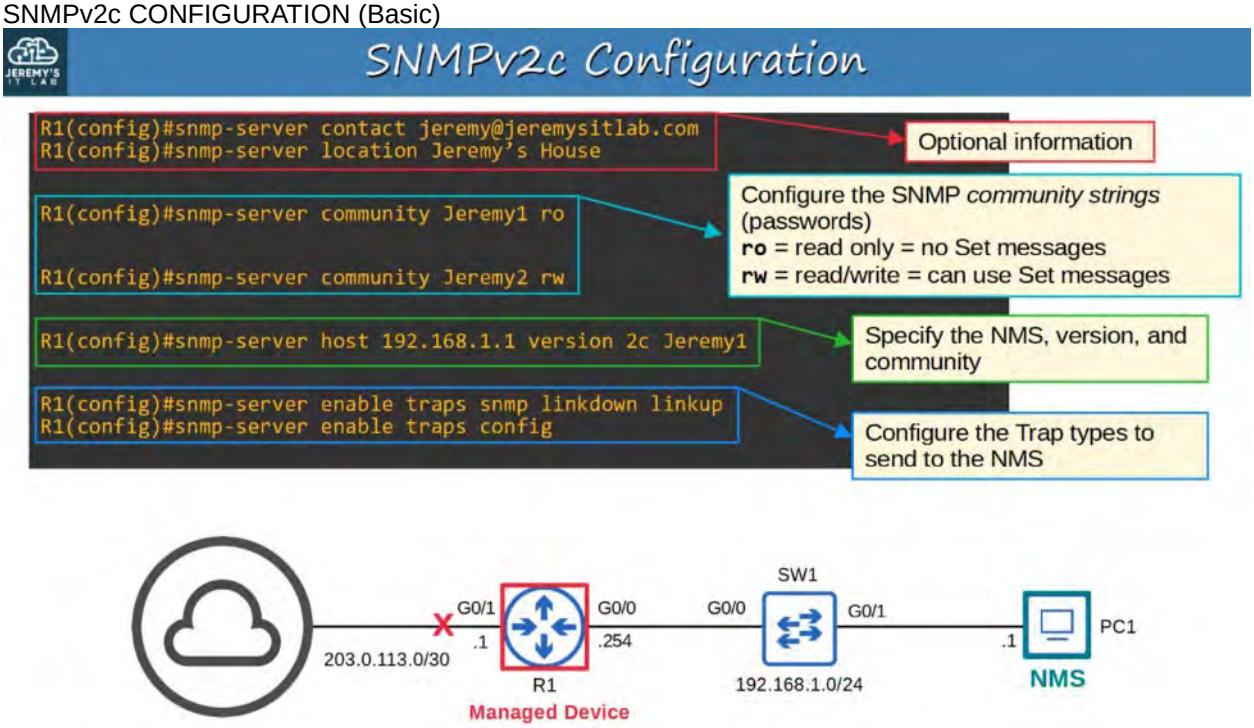
SNMP AGENT listens for MESSAGES on UDP Port 161
SNMP MANAGER listens for MESSAGES on UDP Port 162

SNMP Agents (managed devices) listen for messages on UDP port 161, and SNMP Managers listen for messages on UDP port 162.

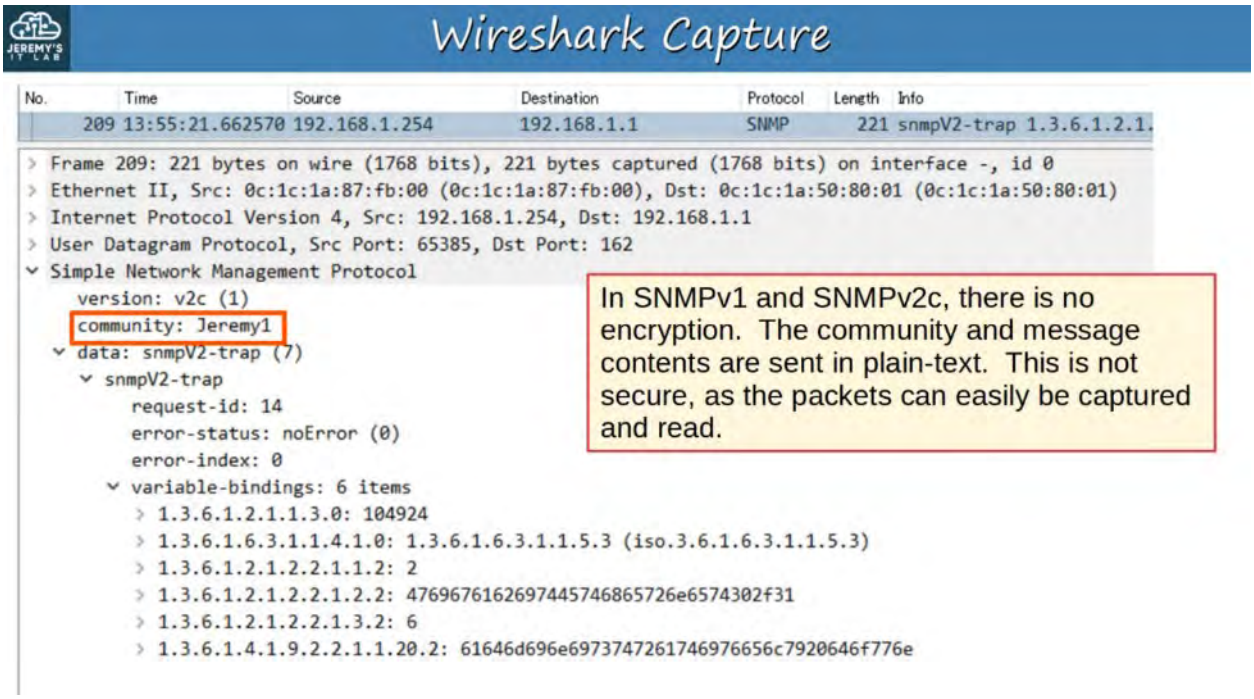
Trap and **Inform** are both messages sent from an SNMP Agent to an SNMP Manager, so they are sent to UDP port 162.

Get and **Set**, on the other hand, are sent from a Manager to an Agent, so they are sent to UDP port 161.

SNMPv2c CONFIGURATION (Basic)



WHAT HAPPENS WITH R1's G0/1 INTERFACE GOES DOWN?



NOTE:
UDP message sent to Destination Port 162 (SNMP Manager)
“version” is set to v2c
community is “Jeremy1” (Read Only - no Set messages)
snmpV2-trap : trap message sent due to interface G0/1 going down
variable-bindings : contains the OID sent to identify the issue.

SNMP SUMMARY

- SNMP helps MANAGE DEVICES over a NETWORK
- MANAGED DEVICES are the devices being managed using SNMP (such as 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, Hostname, etc are STORED in the MANAGMENT INFORMATION BASE (MIB) and identified using Object IDs (OIDs)

Main SNMP versions : SNMPv1, SNMPv2c, SNMPv3
SNMP MESSAGES :
* Get / GetNext / GetBulk
* Set
* Trap
* Inform
* Response

41. SYSLOG

SYSLOG OVERVIEW

- SYSLOG is an INDUSTRY-STANDARD PROTOCOL for message logging
- On NETWORK DEVICES, SYSLOG can be used to LOG EVENTS
  - Changes in INTERFACE status (UP / DOWN)
  - Changes in OSFP NEIGHBOUR 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

seq: time stamp: %facility-severity-MNEMONIC:description

💡 These TWO FIELDS may or may not be displayed, depending on the DEVICE'S configuration

seq = A SEQUENCE NUMBER indicating the order / sequence of messages

time stamp = A TIMESTAMP indicating the time the message was generated

facility = A VALUE that indicates which process on the DEVICE generated the message

severity = A NUMBER that indicates the severity of a logged event.

Official RFC for SYSLOG severity levels

💡 LEVELS and KEYWORDS need to be MEMORIZED for the CCNA

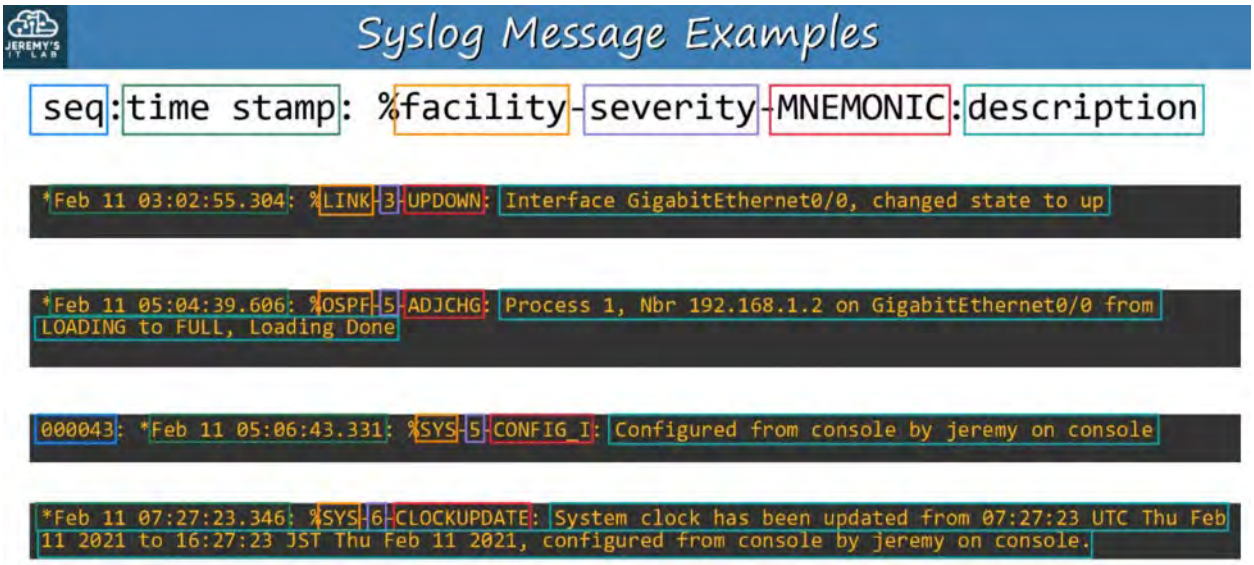
Syslog Severity Levels

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

💡 MEMORIZATION MNEMONIC : (E)very (A)wesome (C)isco (E)ngineer (W)ill (N)eed (I)ce cream (D)aily

MNEMONIC = A SHORT CODE for the message, indicating what happened

description = Detailed information about the EVENT being reported

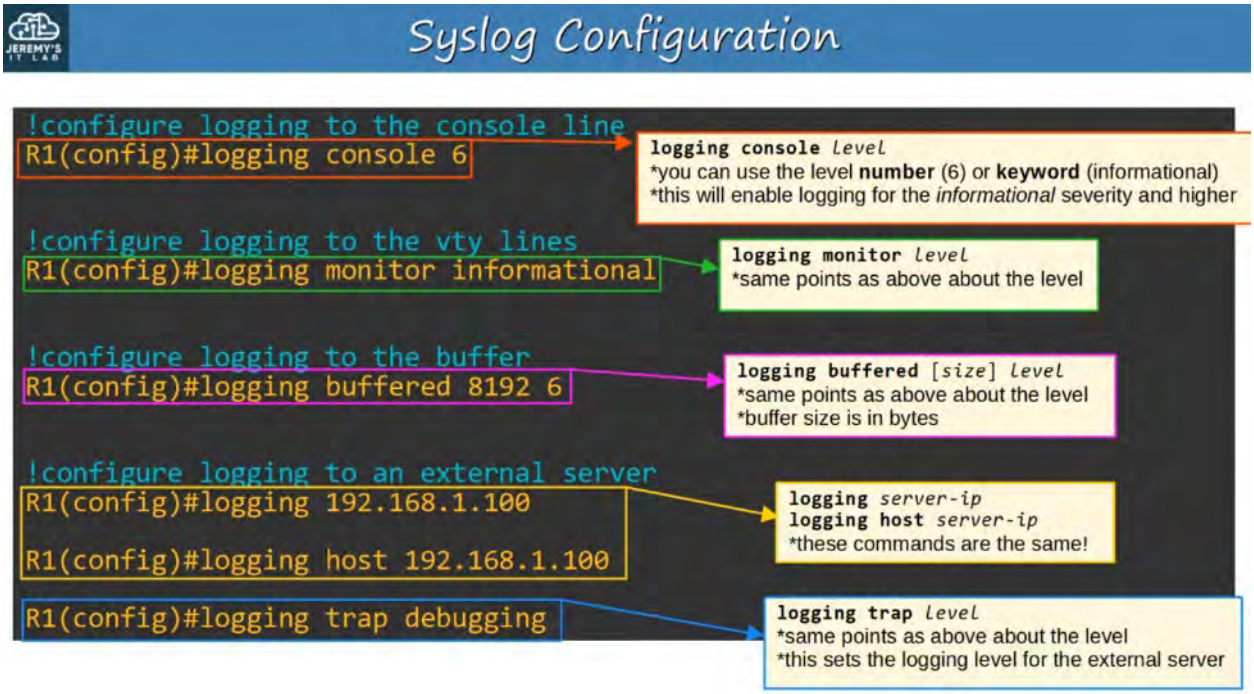


SYSLOG LOGGING LOCATIONS

- **CONSOLE LINE**
  - SYSLOG messages will be displayed in the CLI when connected to the DEVICE via the CONSOLE port. By DEFAULT, all messages (Level 0-7) are displayed
- **BUFFER**
  - Syslog messages will be saved to RAM. By default, ALL messages (Level 0-7) are displayed
- **VTY LINES**
  - SYSLOG messages will be displayed in the CLI when connected to the DEVICE via Telnet/SSH (coming in a later video). Disabled by default.
- **EXTERNAL SERVER**
  - You can configure the DEVICE to send SYSLOG messages to an external server

** SYSLOG SERVERS will listen for messages on UDP PORT 514 **

SYSLOG CONFIGURATION



level works from the chosen level and upward toward Level 0 (EMERGENCY)
level or keyword from the Severity Table works when choosing a level
TERMINAL MONITOR

- Even if logging monitor level is enabled, by default SYSLOG messages will not be displayed when connected via Telnet or SSH
- For the messages to be displayed, you must use the following command:
  - R1# terminal monitor
- The command must be used **every time you connect to the DEVICE via Telnet or SSH**

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
```

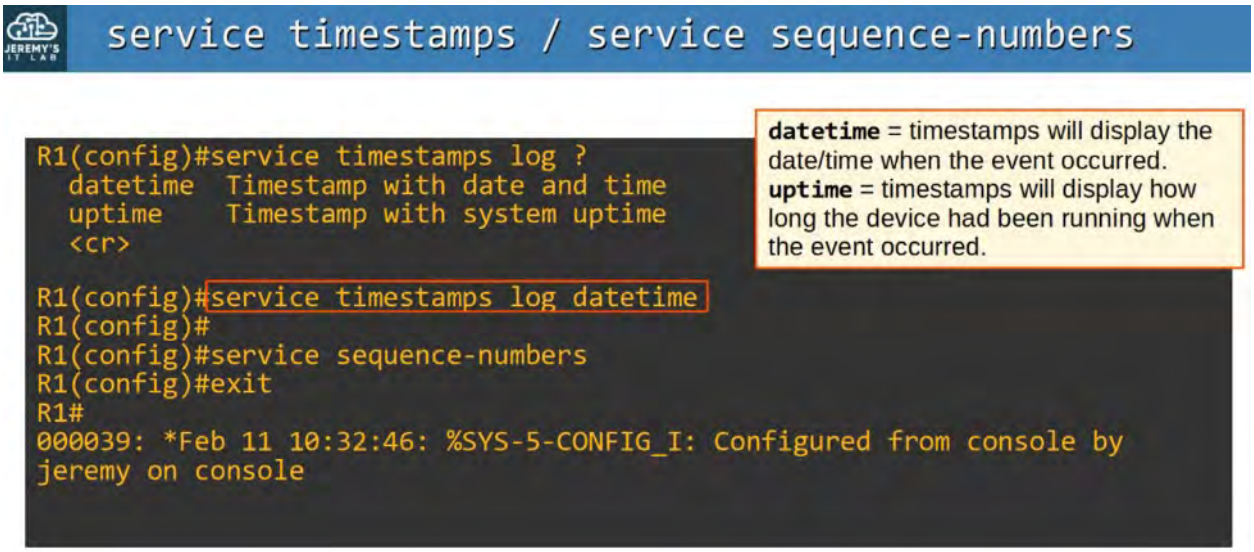
- To prevent this, you should use logging synchronous on the appropriate *line*

```
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
```

SERVICE TIMESTAMPS and SERVICE SEQUENCE-NUMBERS



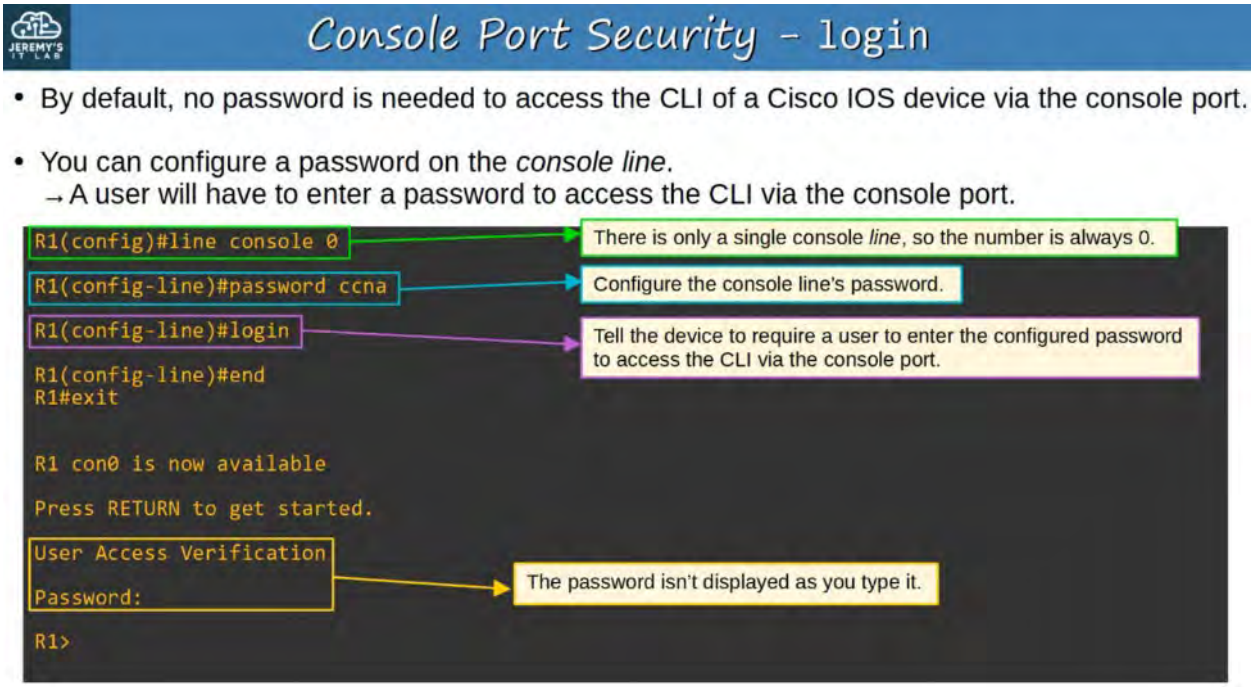
SYSLOG versus SNMP

- SYSLOG and SNMP are both used for MONITORING and TROUBLESHOOTING of DEVICES. They are COMPLIMENTARY, but their FUNCTIONALITIES are different.
- SYSLOG
  - 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
  - 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

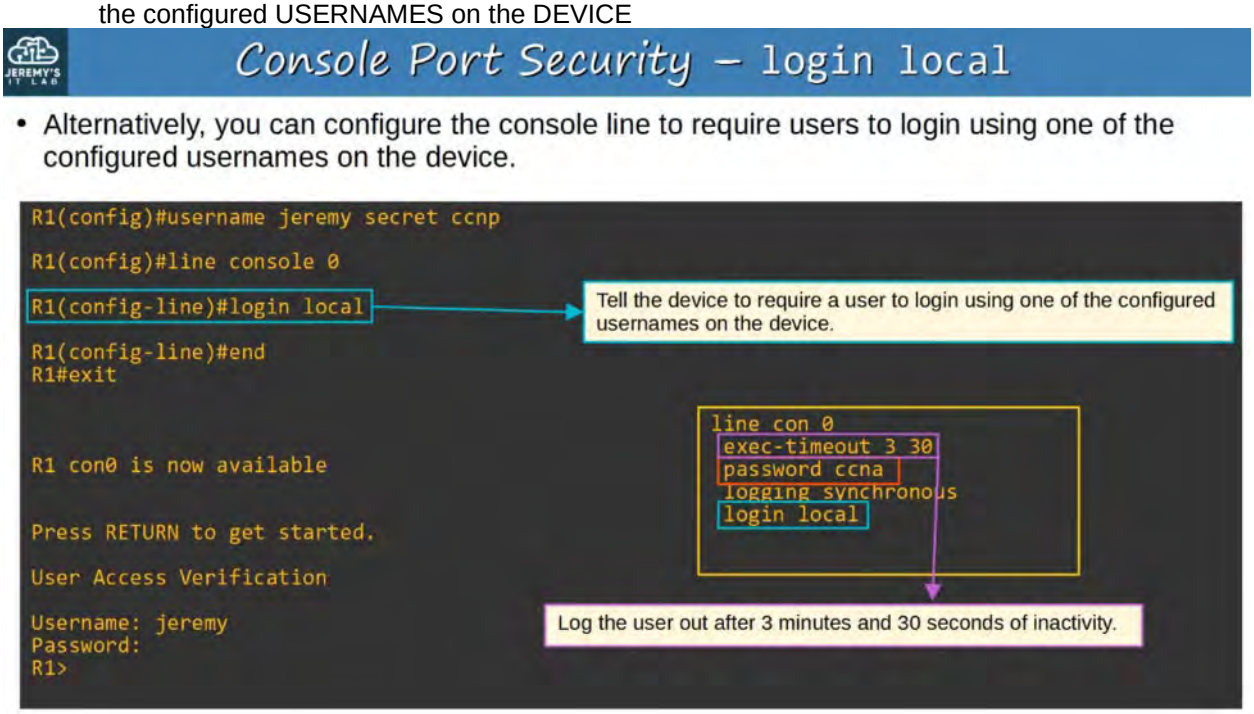
42. SSH (Secure Shell)

CONSOLE PORT SECURITY

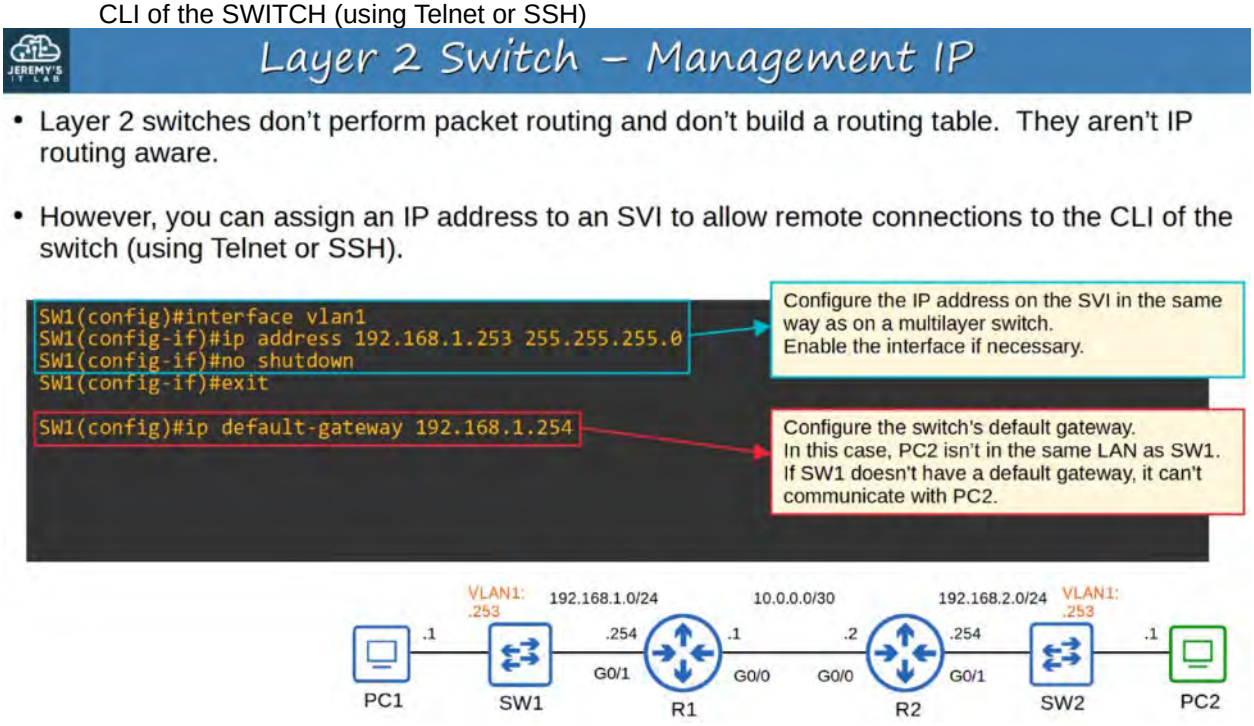
- By DEFAULT, no password us needed to access the CLI of a CISCO IOS DEVICE via the CONSOLE PORT
- You can CONFIGURE a PASSWORD on the *console line*
    - A USER will have to enter a PASSWORD to ACCESS the CLI via the CONSOLE PORT



- Alternatively, you can configure the CONSOLE LINE to require USERS to LOGIN using one of the configured USERNAMES on the DEVICE



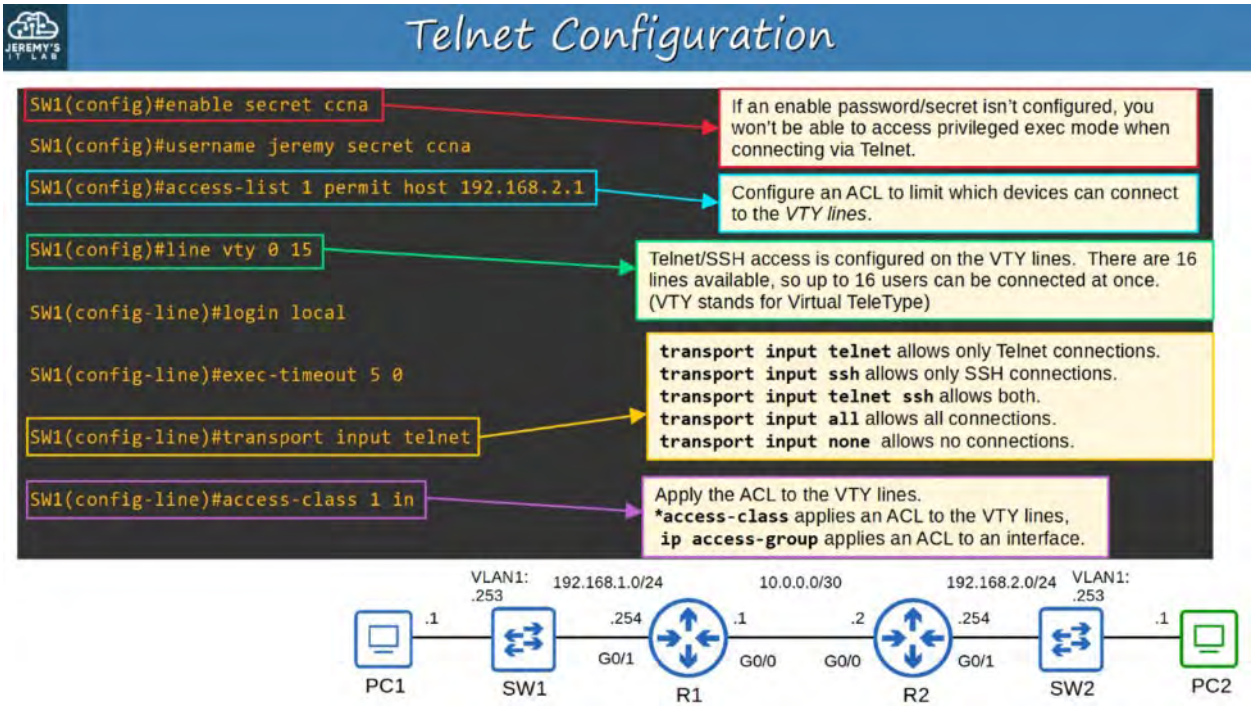
- LAYER 2 SWITCHES do not perform PACKET ROUTING and build a ROUTING TABLE. They are NOT IP ROUTING aware
- However, you CAN assign an IP ADDRESS to an SVI to allow REMOTE CONNECTIONS to the CLI of the SWITCH (using Telnet or SSH)



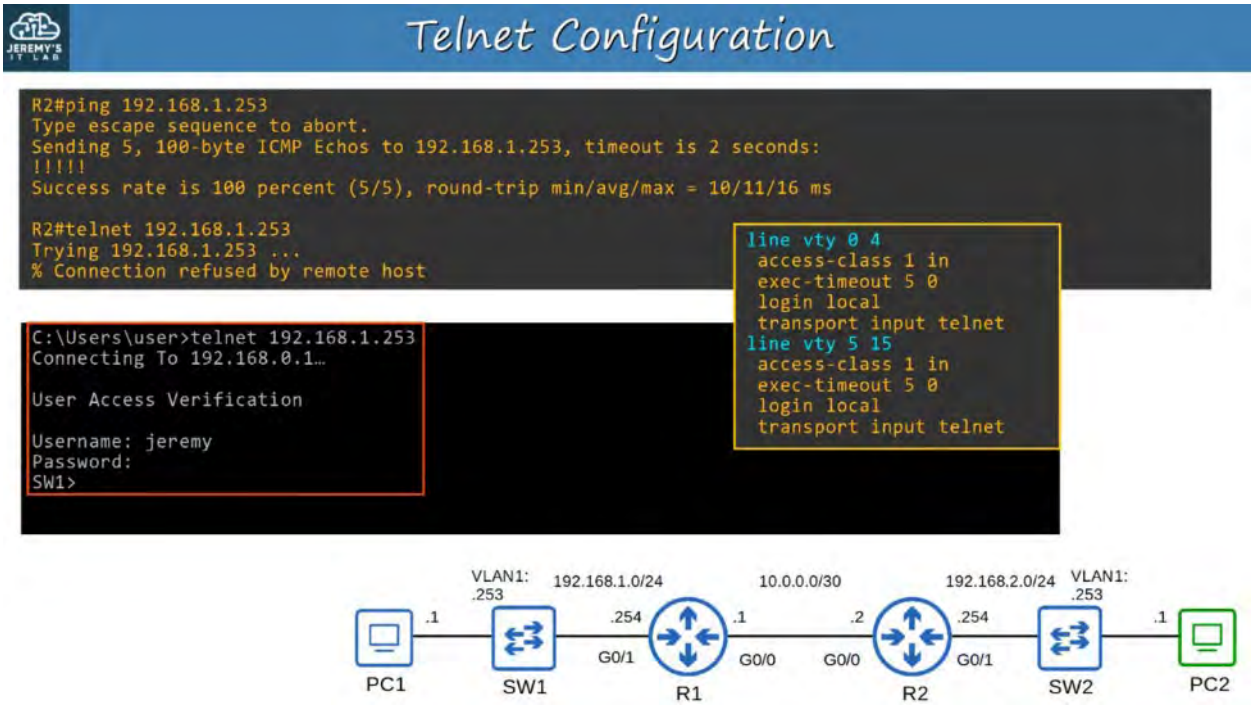
TELNET

- TELNET (Teletype Network) is a PROTOCOL used to REMOTELY ACCESS the CLI of a REMOTE HOST
- TELNET was developed in 1969
- TELNET has been largely REPLACE by SSH, which is MORE Secure
- TELNET sends data in PLAIN TEXT. NO ENCRYPTION(!)

💡 TELNET SERVERS listen for TELNET traffic on TCP PORT 23



VERIFY TELNET CONFIGURATION



SSH

- SSH (Secure Shell) was developed in 1995 to REPLACE LESS SECURE PROTOCOLS, like TELNET
- SSHv2, a major revision of SSHv1, was released in 2006
- If a DEVICE supports both v1 and v2, it is said to run 'version 1.99'
- Provides SECURITY features; such as DATA ENCRYPTION and AUTHENTICATION

CHECK SSH SUPPORT

SSH Configuration: Check SSH Support

```
SW1#show version
Cisco IOS Software, vios_l2 Software (vios_l2-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST
ENGINEERING ESTG_WEEKLY BUILD, synced to  END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Tue 28-Jul-15 18:52 by sasyamal

SW1#show ip ssh
SSH Disabled - version 1.99
%Please create RSA keys to enable SSH (and of atleast 768 bits for SSH v2).
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
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): NONE
```

- IOS images that support SSH will have 'K9' in their name.
- Cisco exports NPE (No Payload Encryption) IOS images to countries that have restrictions on encryption technologies.
- NPE IOS images do not support cryptographic features such as SSH.

RSA KEYS

- To ENABLE and use SSH, you must first generate an RSA PUBLIC and PRIVATE KEY PAIR
- The KEYS are used for DATA ENCRYPTION / DECRYPTION, AUTHENTICATION, etc.

SSH Configuration: RSA Keys

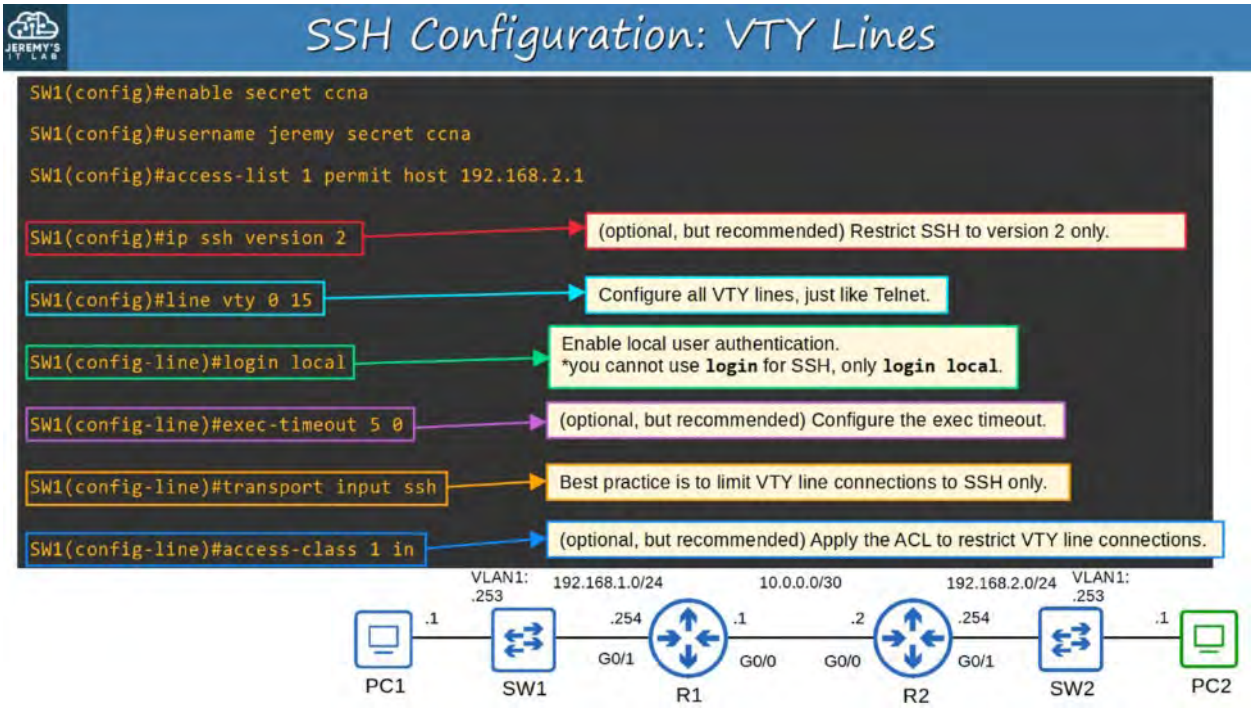
- To enable and use SSH, you must generate an RSA public and private key pair.
- The keys are used for data encryption/decryption, authentication, etc.

```
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]
```

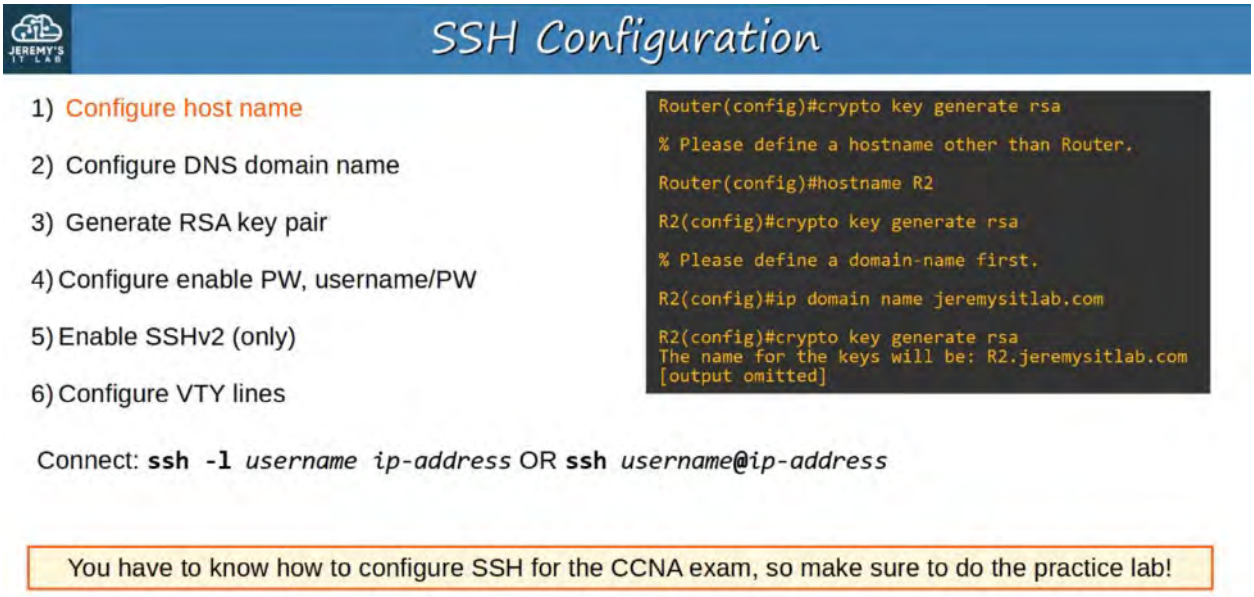
The **FQDN** of the device is used to name the RSA keys.
FQDN = Fully Qualified Domain Name (host name + domain name)

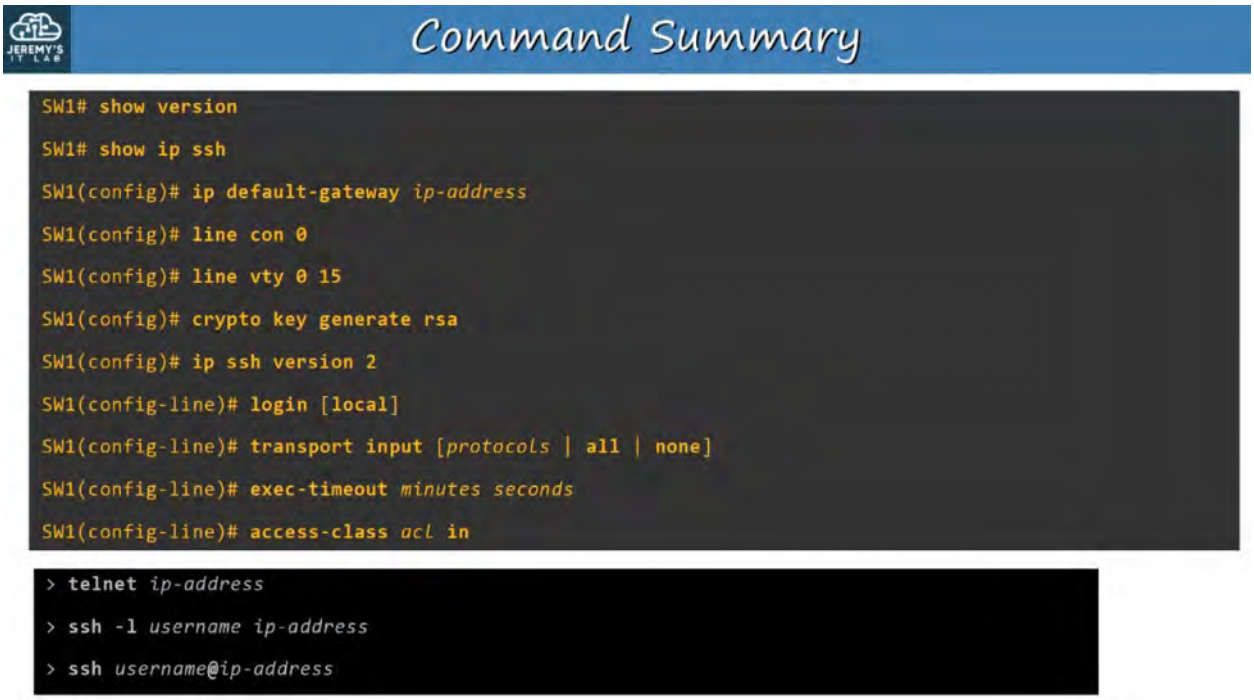
Generate the RSA keys.
crypto key generate rsa modulus *Length*
is an alternate method.
*length must be 768 bits or greater for SSHv2

VTY LINES



SUMMARY ABOUT SSH CONFIGURATIONS

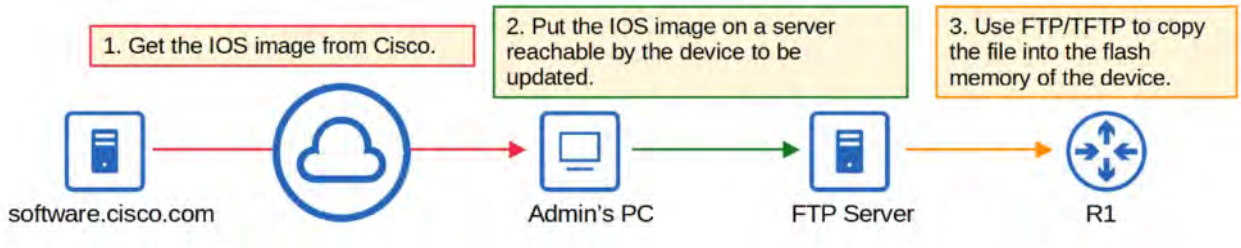



Command Summary
SW1# show version
SW1# show ip ssh
SW1(config)# ip default-gateway ip-address
SW1(config)# line con 0
SW1(config)# line vty 0 15
SW1(config)# crypto key generate rsa
SW1(config)# ip ssh version 2
SW1(config-line)# login [local]
SW1(config-line)# transport input [protocols | all | none]
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

43. FTP and TFTP

THE PURPOSE OF 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 / TFTP to COPY files FROM a SERVER
    - CLIENTS can use FTP / 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



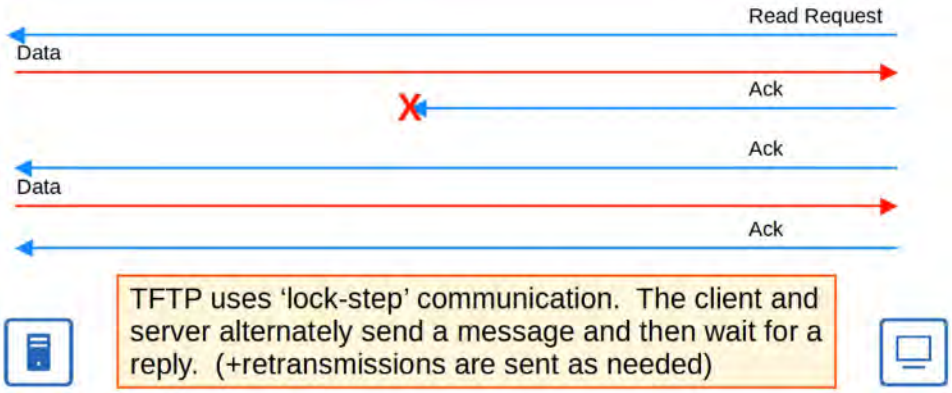
TFTP and FTP FUNCTIONS AND DIFFERENCES

TFTP

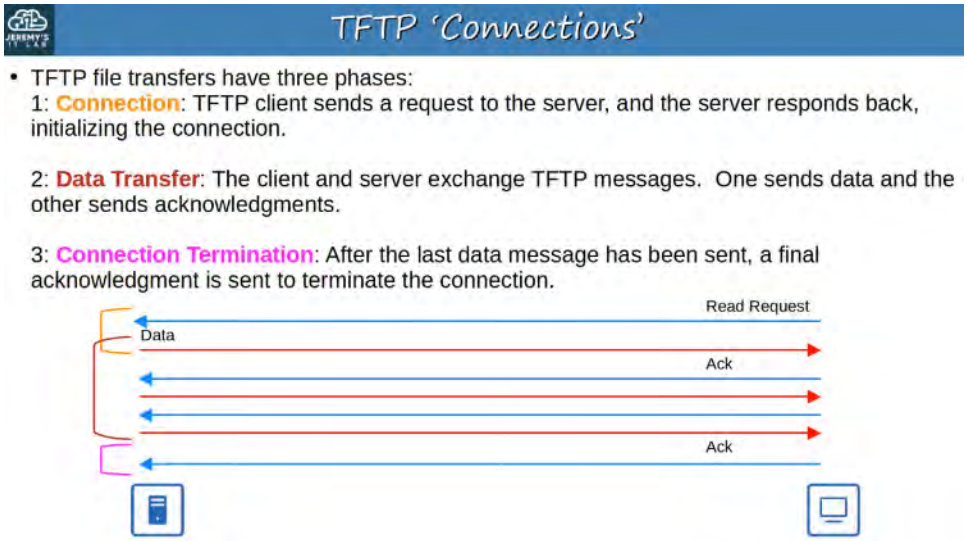
- TFTP first standardized in 1981
- Named "Trivial" because it's SIMPLE and has only basic features compared to FTP
    - Only allows a CLIENT to COPY FILES to / from a SERVER
- Was released after FTP, but not a REPLACEMENT for FTP.
    - It's another tool to use when LIGHTWEIGHT SIMPLICITY is more important than FUNCTIONALITY
- NO AUTHENTICATION (Username / Password) so SERVERS will respond to ALL FTP REQUESTS
- NO ENCRYPTION. All DATA is sent 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 provided 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 RESEND its previous message.



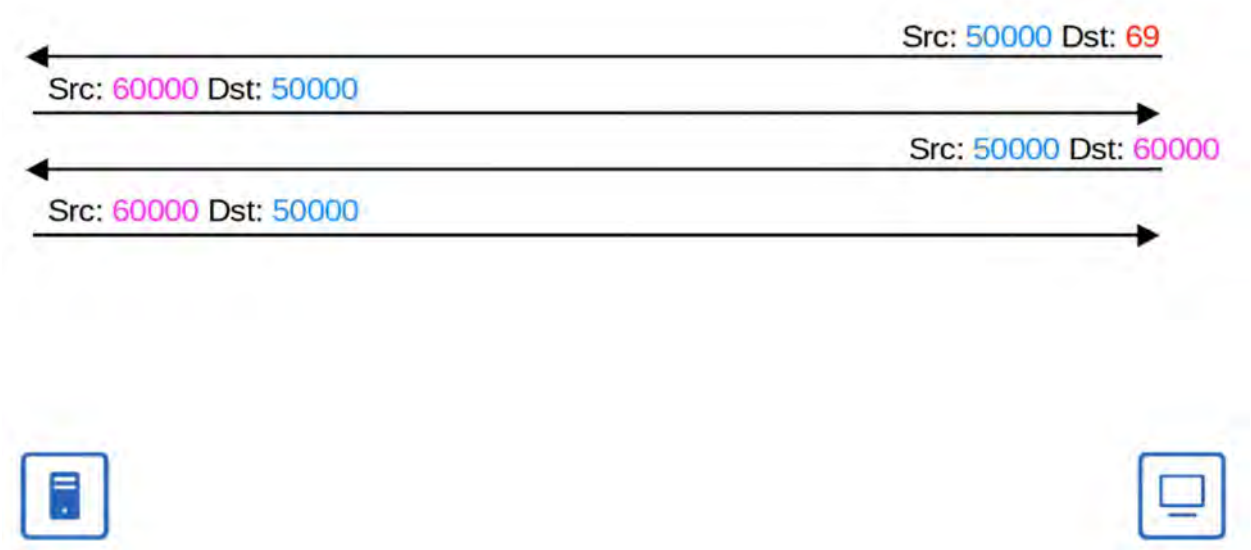
TFTP "CONNECTIONS"



TFTP TID (Not in the CCNA exam)

- When the CLIENT sends the FIRST message to the SERVER, the DESTINATION PORT is UDP 69 and the SOURCE PORT is a random EPHEMERAL PORT
- This "random port" is called a "TRANSFER IDENTIFIER" (TID) and identifies the DATA TRANSFER
- The SERVER then also selects a RANDOM TID to use as a SOURCE PORT when it replies, NOT UDP 69
- When the CLIENT sends the NEXT message, the DESTINATION PORT will be the SERVER'S TID, NOT UDP 69

UDP PORT 69 (TFTP) is only used at the initial request message

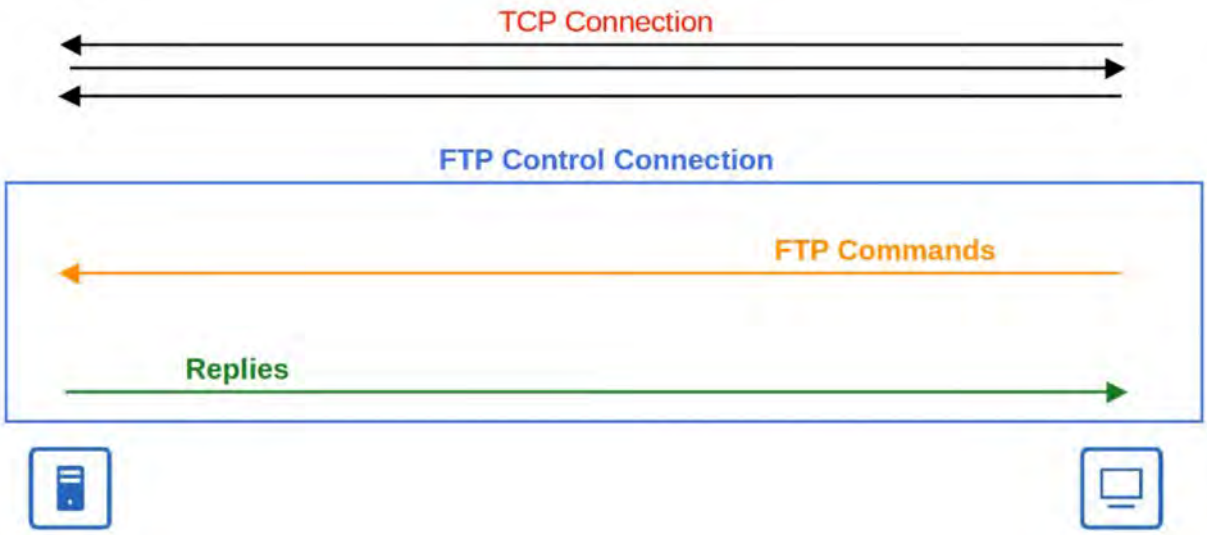


FTP

- 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 / REMOVE FILES
    - LIST FILES
    - etc…
- The CLIENT sends FTP *commands* to the SERVER to perform these functions

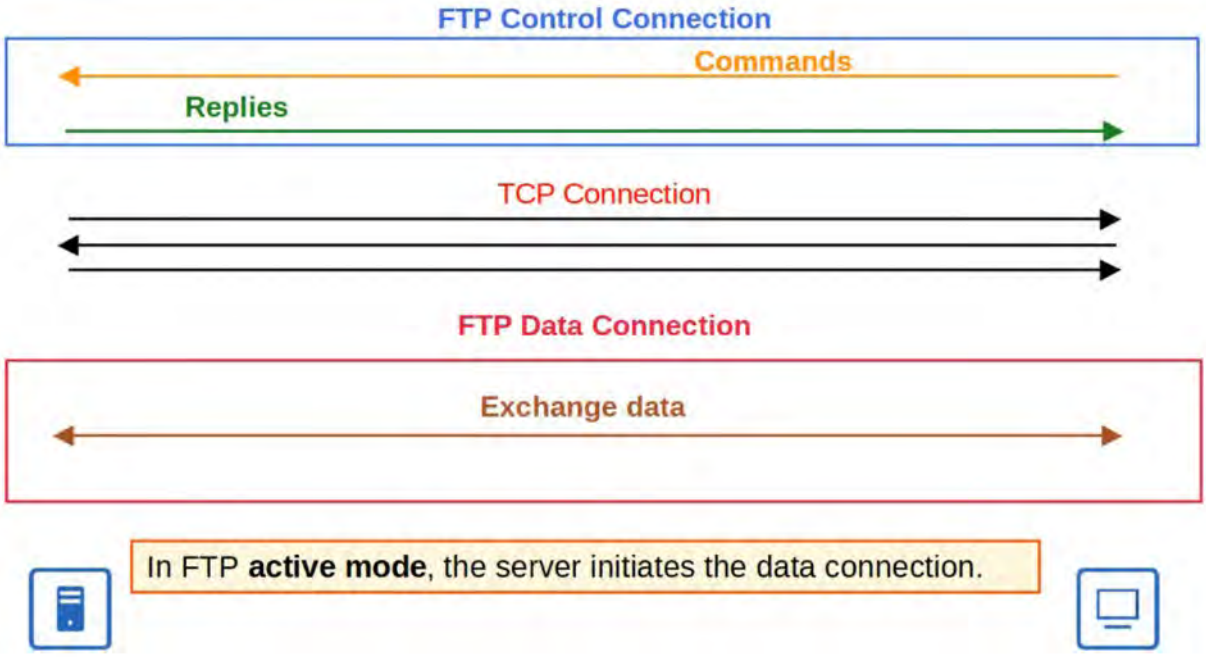
FTP CONTROL CONNECTIONS

- 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

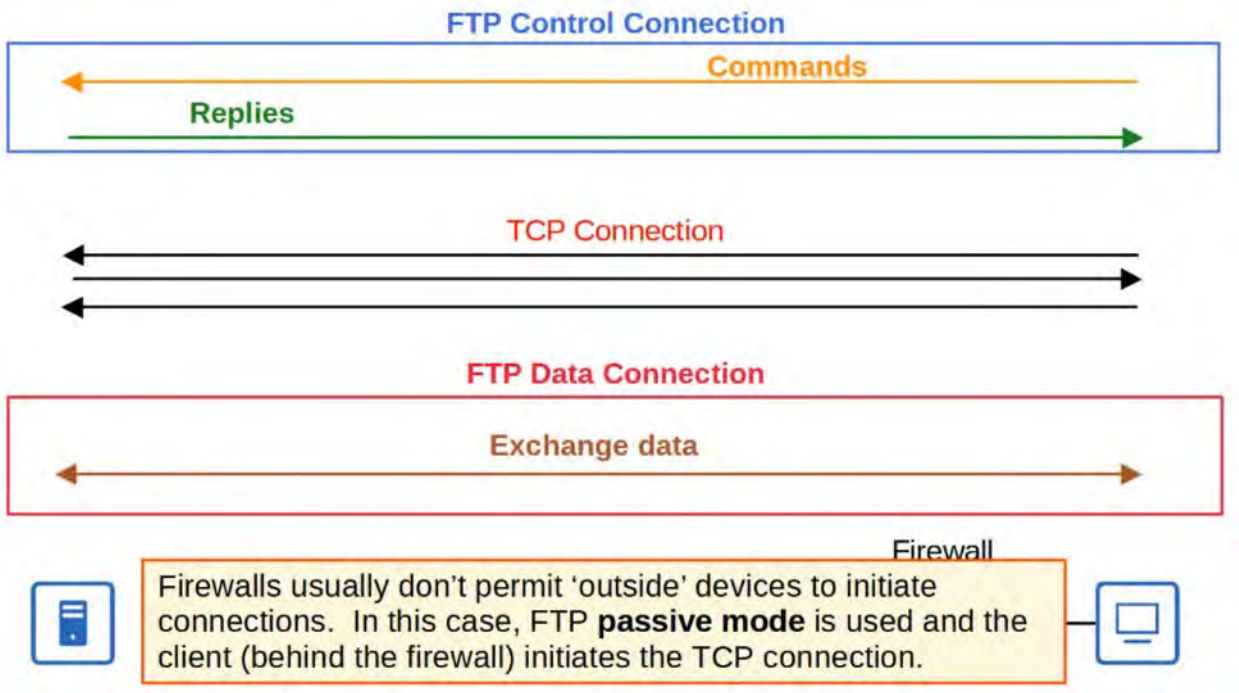


ACTIVE MODE FTP DATA CONNECTIONS

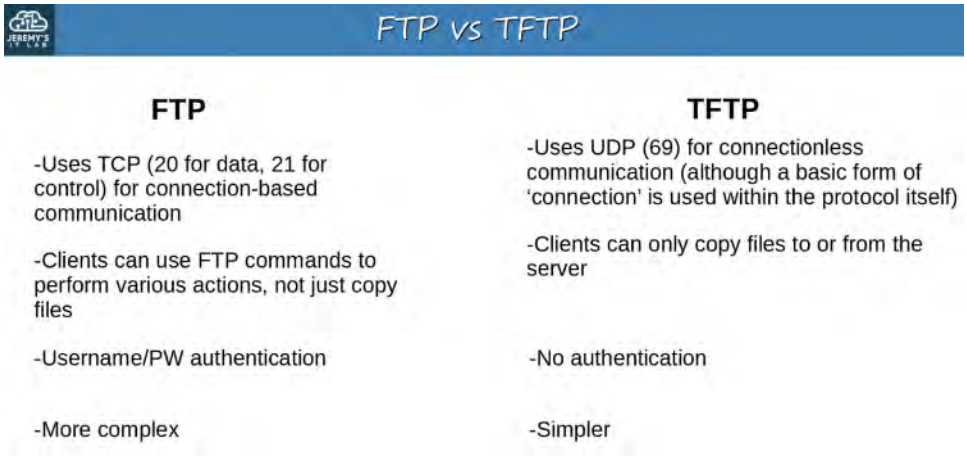
- The DEFAULT method of establishing FTP DATA connections is ACTIVE MODE in which the SERVER initiates the TCP connection.



- 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



FTP VERSUS TFTP



IOS FILE SYSTEMS

- A FILE SYSTEM is a way of controlling how DATA is STORED and RETRIEVED
- You can VIEW the FILE SYSTEM of a Cisco IOS DEVICE with show file systems

IOS File Systems

- A file system is a way of controlling how data is stored and retrieved.
- You can view the file systems of a Cisco IOS device with **show file systems**

```
Router#show file systems
File Systems:

      Size(b)       Free(b)      Type  Flags  Prefixes
*   2142715904    1994403840      disk     rw   flash0: flash:#
            -             -      disk     rw   flash1:
       966656        962560      disk     rw   flash2:#
            -             -      disk     rw   flash3:
            -             -    opaque     rw   archive:
            -             -    opaque     rw   system:
       262144        256791     nvram     rw   nvram:
            -             -    opaque     rw   tmpsys:
            -             -   network     rw   snmp:
            -             -    opaque     rw   null:
            -             -   network     rw   tftp:
            -             -    opaque     ro   xmodem:
            -             -    opaque     ro   ymodem:
            -             -    opaque     wo   syslog:
            -             -   network     rw   rcp:
            -             -   network     rw   pram:
            -             -   network     rw   ftp:
[output omitted]
```

disk: Storage devices such as flash memory.

opaque: Used for internal functions

nvram: Internal NVRAM. The startup-config file is stored here.

network: Represents external file systems, for example external FTP/TFTP servers.

USING FTP / TFTP IN IOS

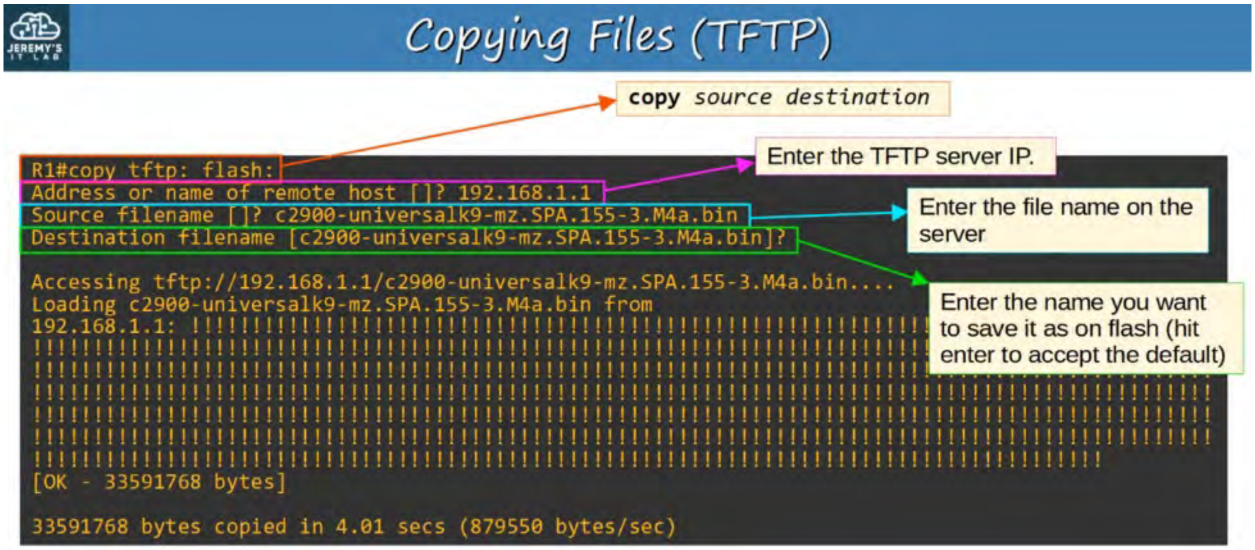
- You can VIEW the current version of IOS with show version

```
R1#show version
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.1(4)M4, RELEASE SOFTWARE
(fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thurs 5-Jan-12 15:41 by pt_team
[output omitted]
```

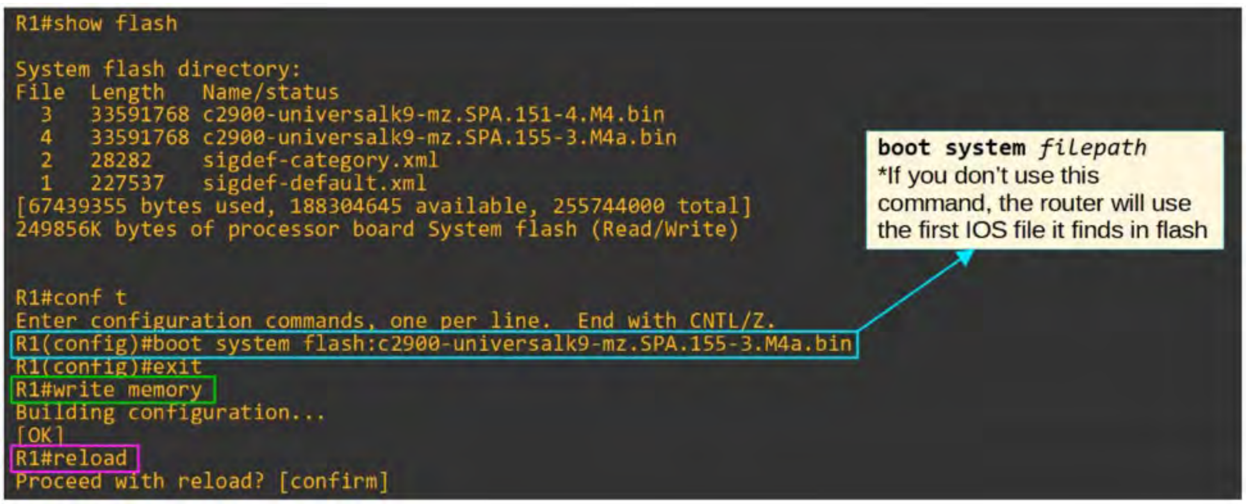
- You can VIEW the contents of flash with show flash

```
R1#show flash
System flash directory:
File  Length   Name/status
  3   33591768 c2900-universalk9-mz.SPA.151-4.M4.bin
  2   28282    sigdef-category.xml
  1   227537   sigdef-default.xml
[33847587 bytes used, 221896413 available, 255744000 total]
249856K bytes of processor board System flash (Read/Write)
```

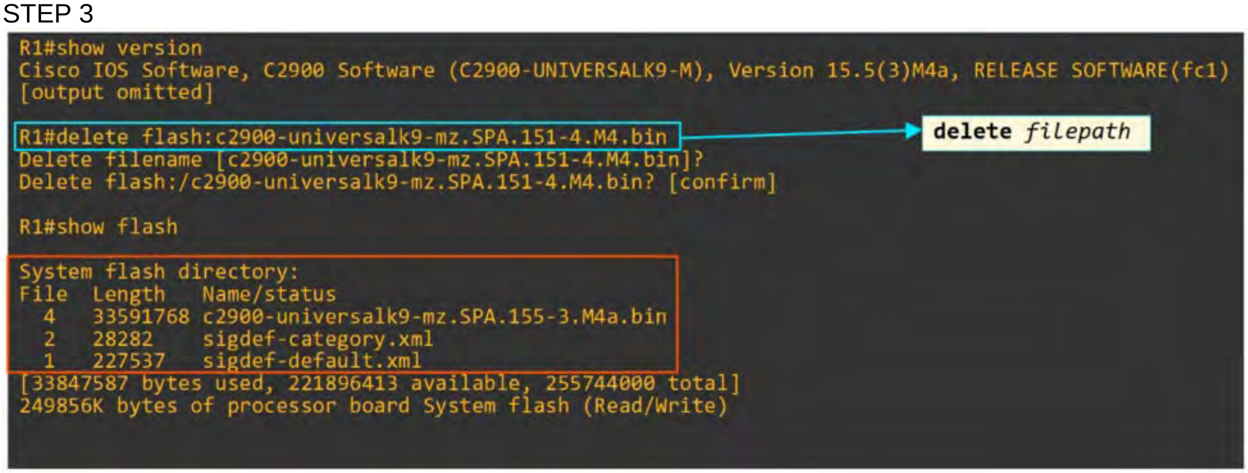
COPYING FILES WITH TFTP
STEP 1



STEP 2

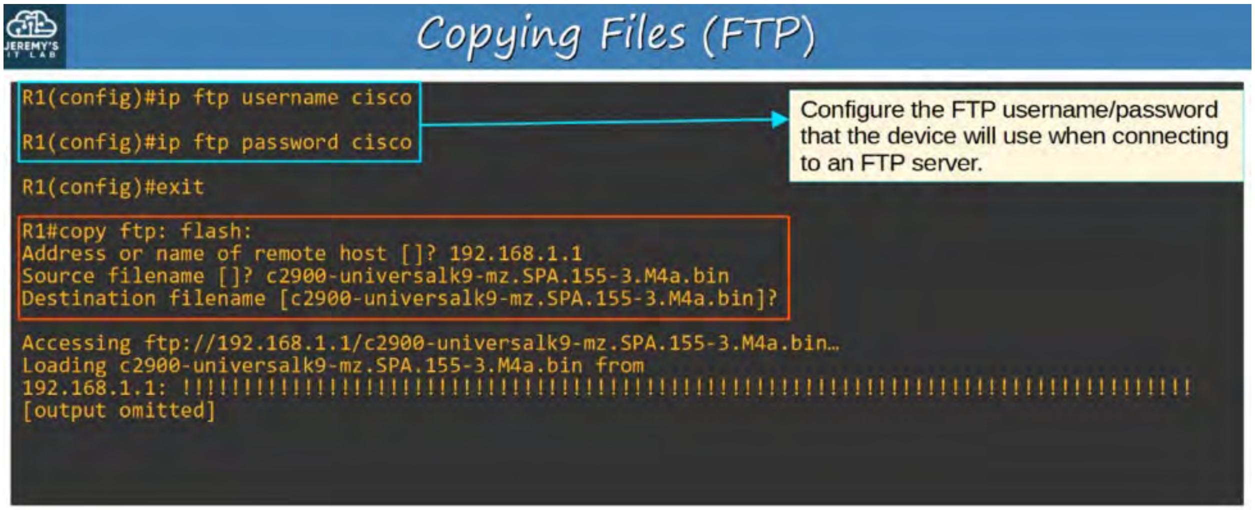


STEP 3



COPYING FILES WITH FTP

STEP 1



STEP 2 and 3 identical to TFTP above

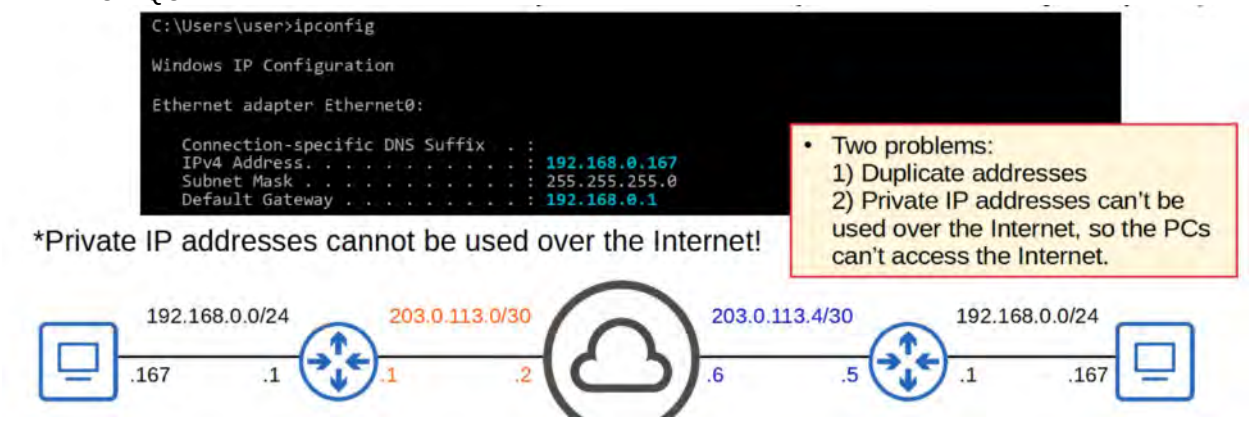
COMMAND SUMMARY

```
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
```

44. NAT (STATIC): PART 1

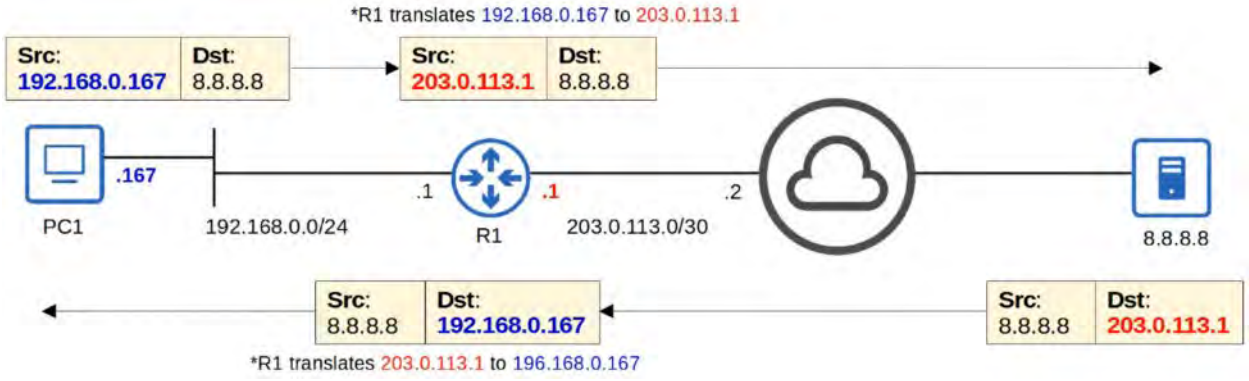
PRIVATE IPv4 ADDRESSES (RFC 1918)

- 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:
  - CIDR
  - PRIVATE IPv4 ADDRESS
  - 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
- You are free to use these ADDRESSES in your NETWORKS. They don't have to be GLOBALLY UNIQUE



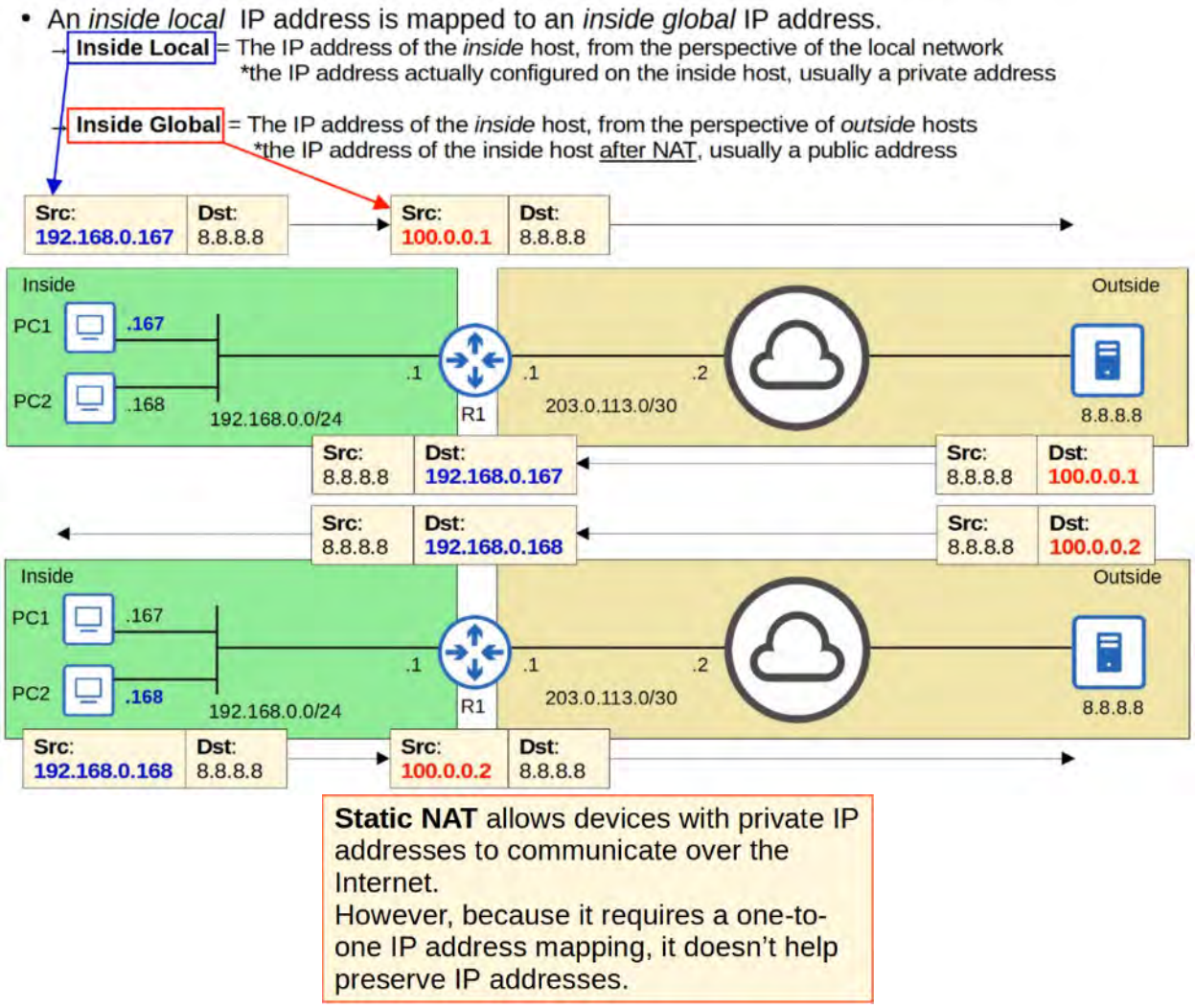
INTRO TO NAT

- 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 ADDRESSES



Static NAT allows devices with private IP addresses to communicate over the Internet.
However, because it requires a one-to-one IP address mapping, it doesn't help preserve IP addresses.

PRIVATE IP CANNOT BE MAPPED TO THE SAME GLOBAL IP
THE SECOND MAPPING WILL BE REJECTED

```
R1(config)#ip nat inside source static 10.0.0.1 20.0.0.1
R1(config)#ip nat inside source static 10.0.0.2 20.0.0.1
% similar static entry (10.0.0.1 -> 20.0.0.1) already exists
```

STATIC NAT CONFIGURATIONS

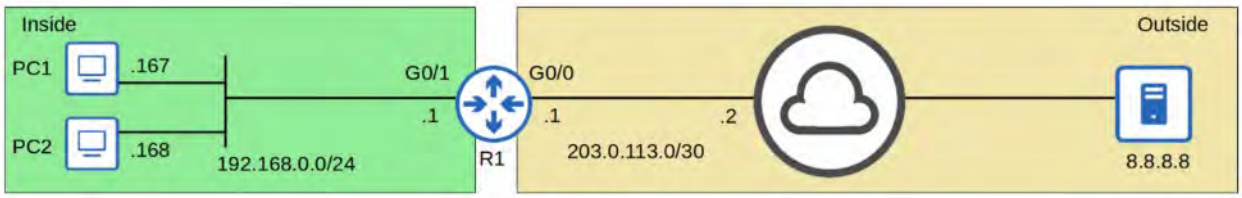
Static NAT Configuration

```
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.

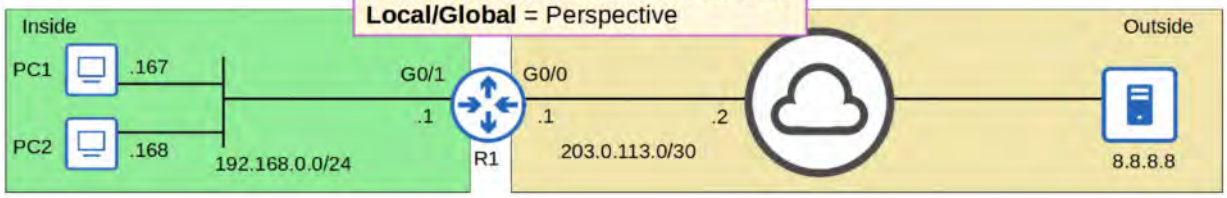
→ **Inside Local** = The IP address of the *inside* host, from the perspective of the local network
*the IP address actually configured on the inside host, usually a private address

→ **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

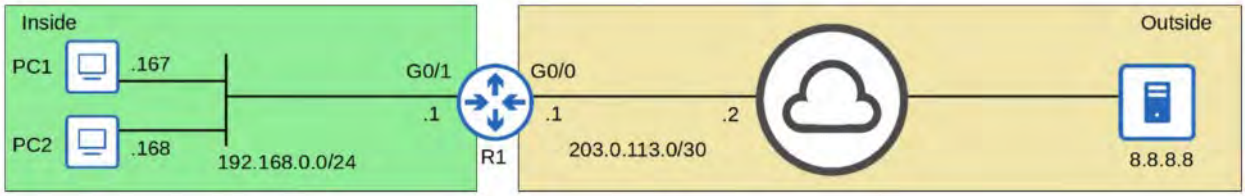
Inside/Outside = Location of the host
Local/Global = Perspective



Command clear ip nat translation

clear ip nat translation *

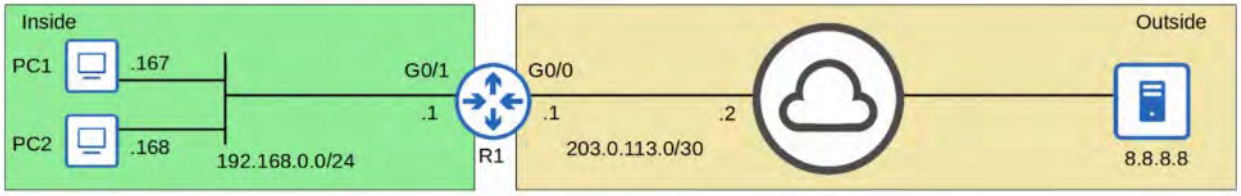
```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 100.0.0.1:56310    192.168.0.167:56310 8.8.8.8:53         8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                 ---
udp 100.0.0.2:62321    192.168.0.168:62321 8.8.8.8:53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

R1#clear ip nat translation *

R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 100.0.0.1          192.168.0.167      ---                 ---
--- 100.0.0.2          192.168.0.168      ---                 ---
```



Command show ip nat statistics

show ip nat statistics

```
R1#show ip nat statistics
Total active translations: 2 (2 static, 0 dynamic; 0 extended)
Peak translations: 4, occurred 02:29:00 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 34  Misses: 0
CEF Translated packets: 30, CEF Punted packets: 4
Expired translations: 4
Dynamic mappings:

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```



COMMAND REVIEW

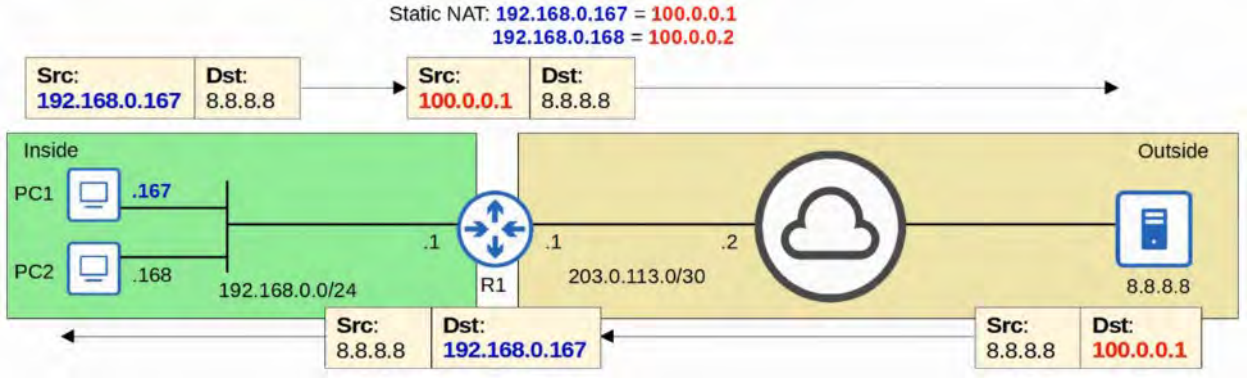
Command Review

```
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 *
```

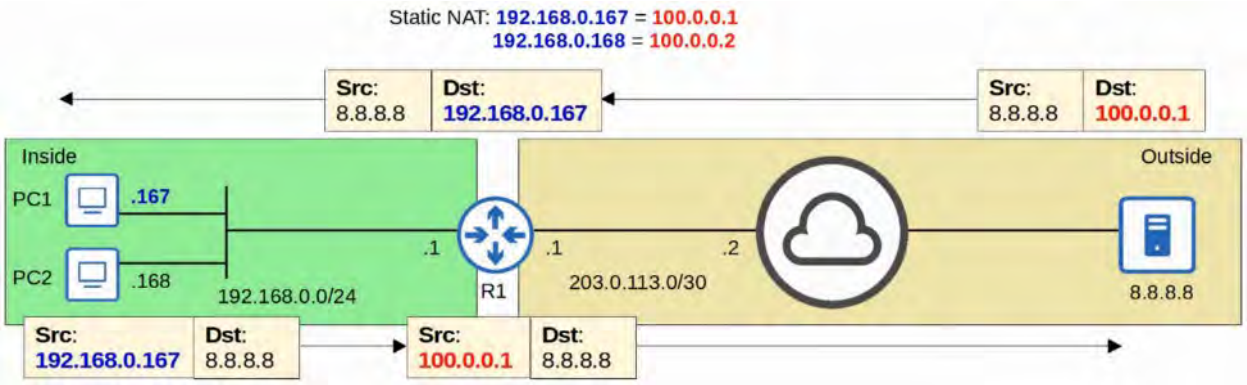
45. NAT (DYNAMIC): PART 2

MORE ABOUT 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



- HOWEVER, this one-to-one mapping also allows EXTERNAL HOSTS to access the INTERNAL HOST via INSIDE GLOBAL ADDRESS

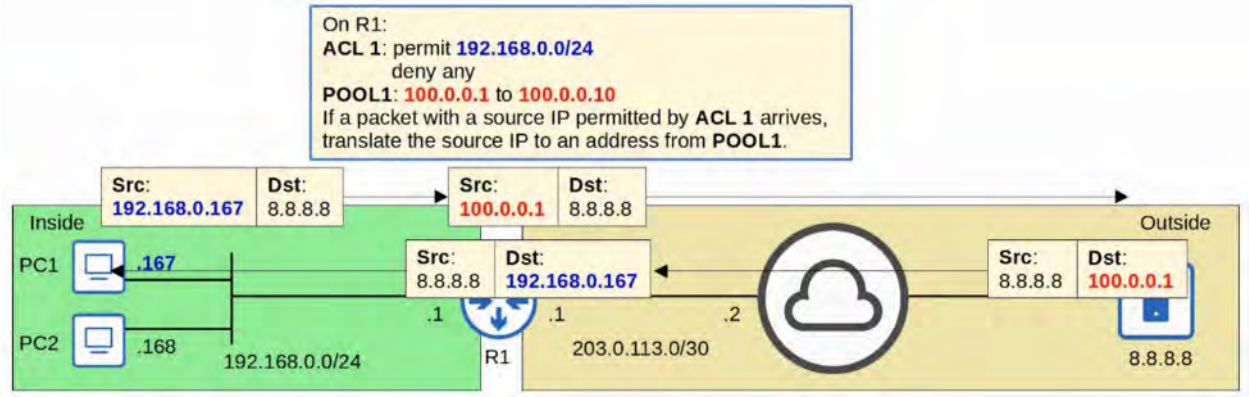


DYNAMIC NAT

- 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; the SOURCE IP will be translated
  - If the SOURCE IP is DENIED; the SOURCE IP will NOT be translated

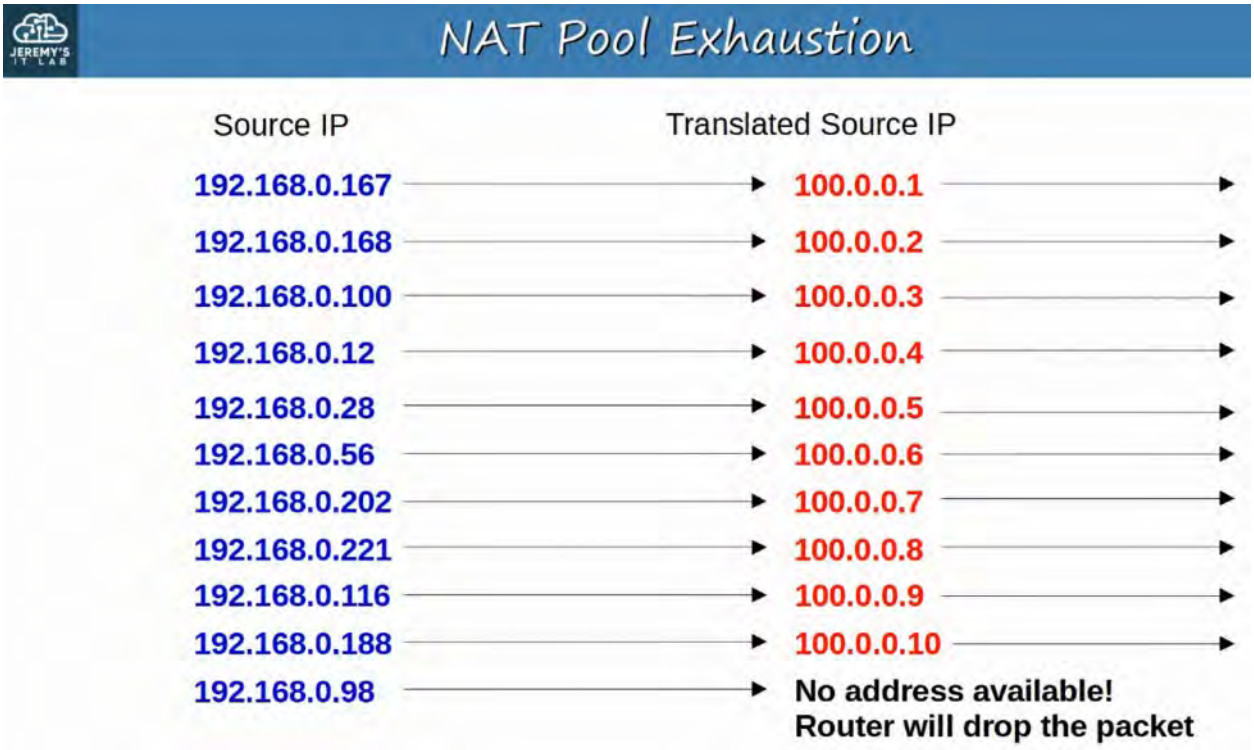
💡 However, Packet Traffic will NOT be dropped

- A NAT POOL is used to define the available INSIDE GLOBAL ADDRESS

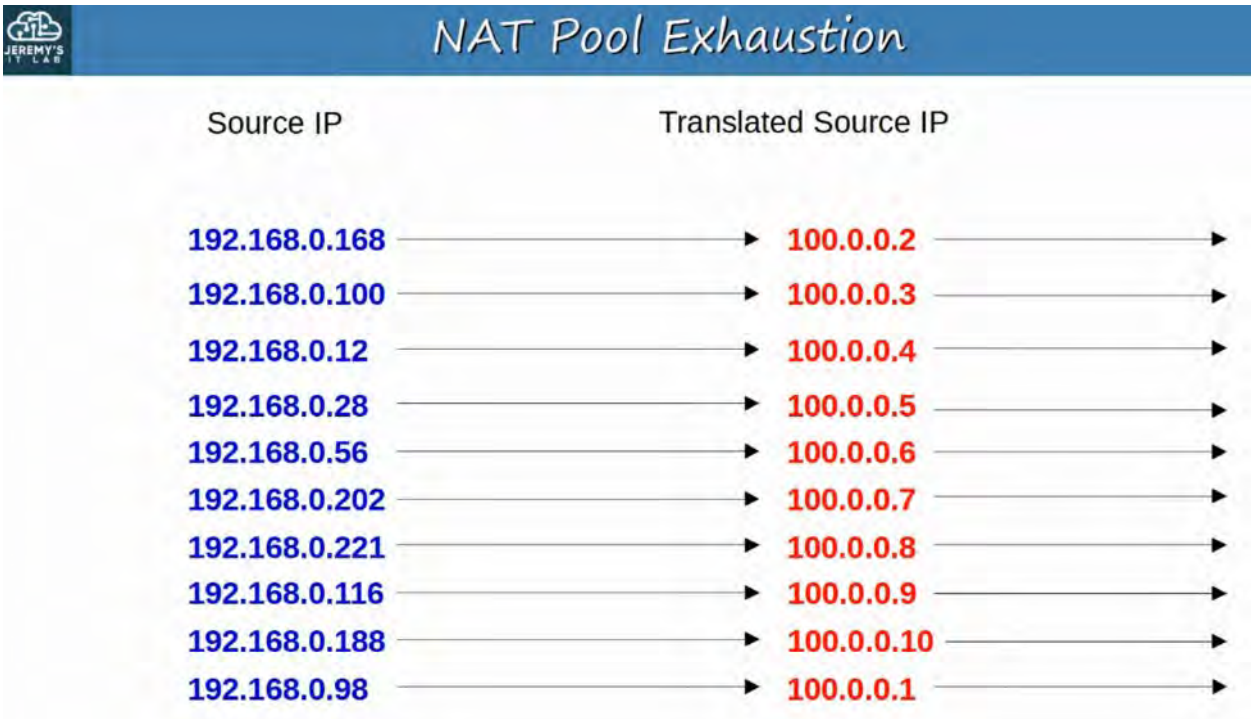


- Although they are dynamically assigned, the mappings are still one-to-one (one INSIDE LOCAL IP ADDRESS per INSIDE GLOBAL IP ADDRESS)
- If there are NOT enough INSIDE GLOBAL IP ADDRESSES available (=ALL are being used), it is called 'NAT POOL EXHAUSTION'
    - If a PACKET from another INSIDE HOST arrives 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

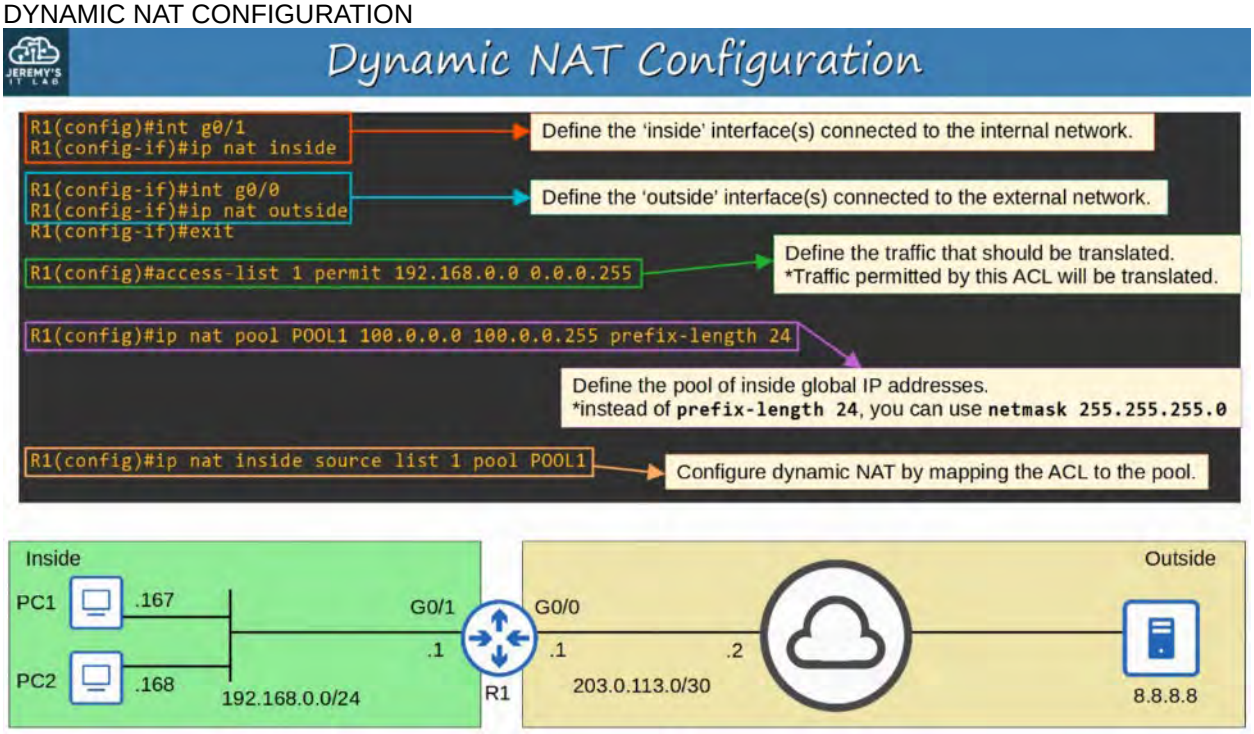
NAT POOL EXHAUSTION



192.168.0.167 TIMES OUT and 192.168.0.98 is assigned it's TRANSLATED SOURCE IP



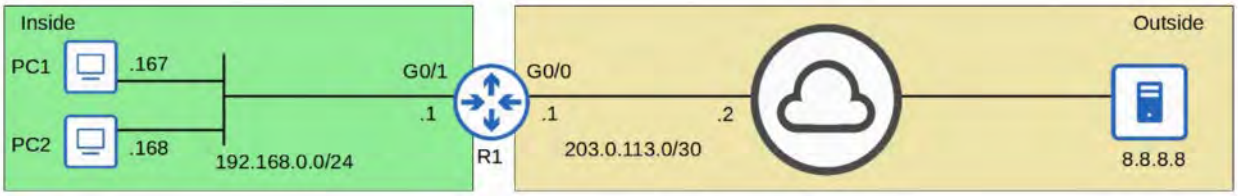
DYNAMIC NAT CONFIGURATION



show ip nat translations

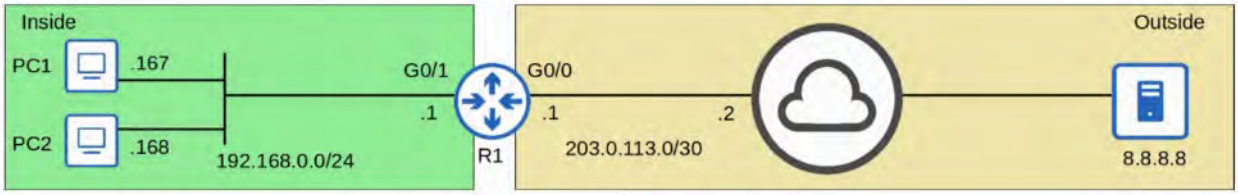
Dynamic NAT Configuration

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 100.0.0.1:3       192.168.0.167:3    8.8.8.8:3          8.8.8.8:3
udp 100.0.0.1:58685    192.168.0.167:58685 8.8.8.8:53        8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                ---
icmp 100.0.0.2:3       192.168.0.168:3    8.8.8.8:3          8.8.8.8:3
udp 100.0.0.2:49536    192.168.0.168:49536 8.8.8.8:53        8.8.8.8:53
--- 100.0.0.2          192.168.0.168      ---                ---
```



Dynamic NAT Configuration

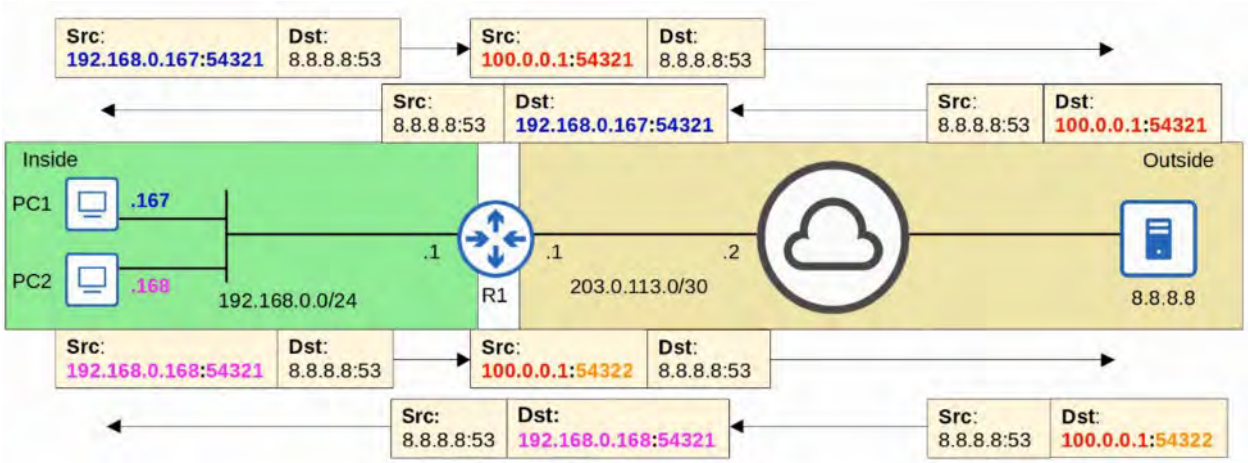
```
R1#show ip nat statistics
Total active translations: 6 (0 static, 6 dynamic; 4 extended)
Peak translations: 6, occurred 00:00:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 32  Misses: 0
CEF Translated packets: 20, CEF Punted packets: 12
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool POOL1 refcount 6
 pool POOL1: netmask 255.255.255.0
        start 100.0.0.0 end 100.0.0.255
        type generic, total addresses 256, allocated 2 (0%), misses 0
[output omitted]
```



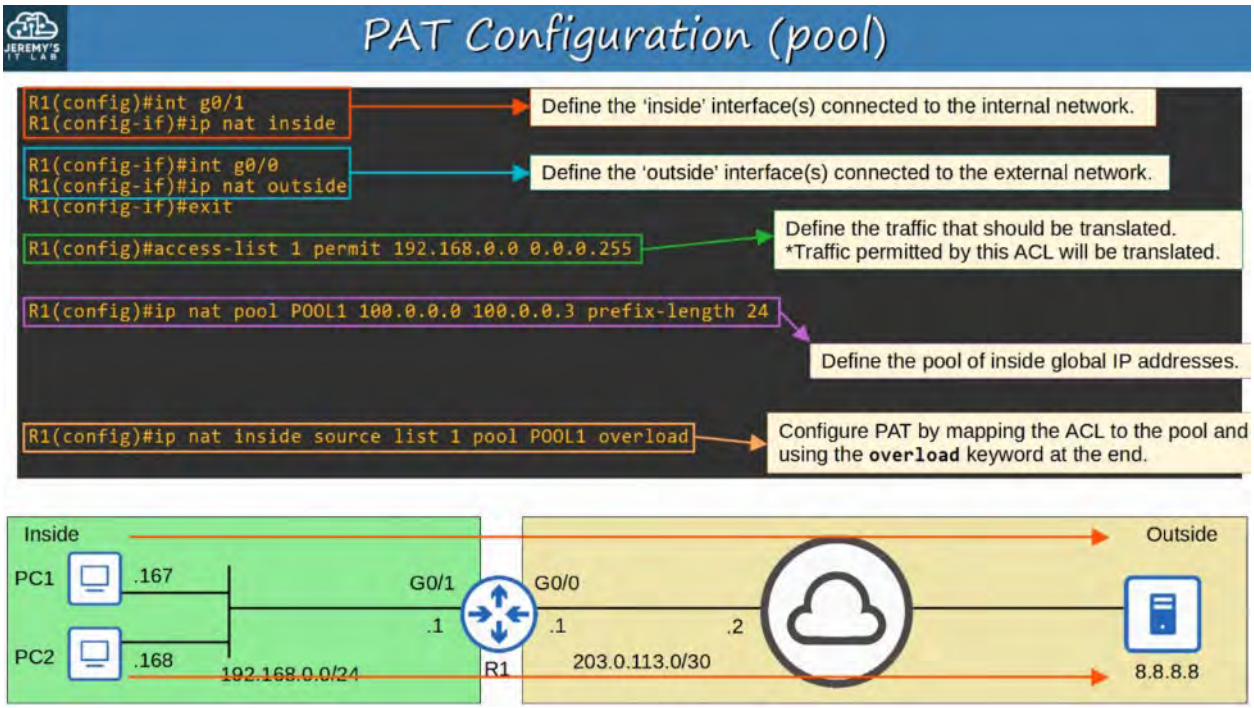
```
R1#show ip nat statistics
Total active translations: 6 (0 static, 6 dynamic; 4 extended)
Peak translations: 6, occurred 00:00:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 32  Misses: 0
CEF Translated packets: 20, CEF Punted packets: 12
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool POOL1 refcount 6
 pool POOL1: netmask 255.255.255.0
        start 100.0.0.0 end 100.0.0.255
        type generic, total addresses 256, allocated 2 (0%), misses 0
[output omitted]
```

DYNAMIC PAT (NAT OVERLOAD)

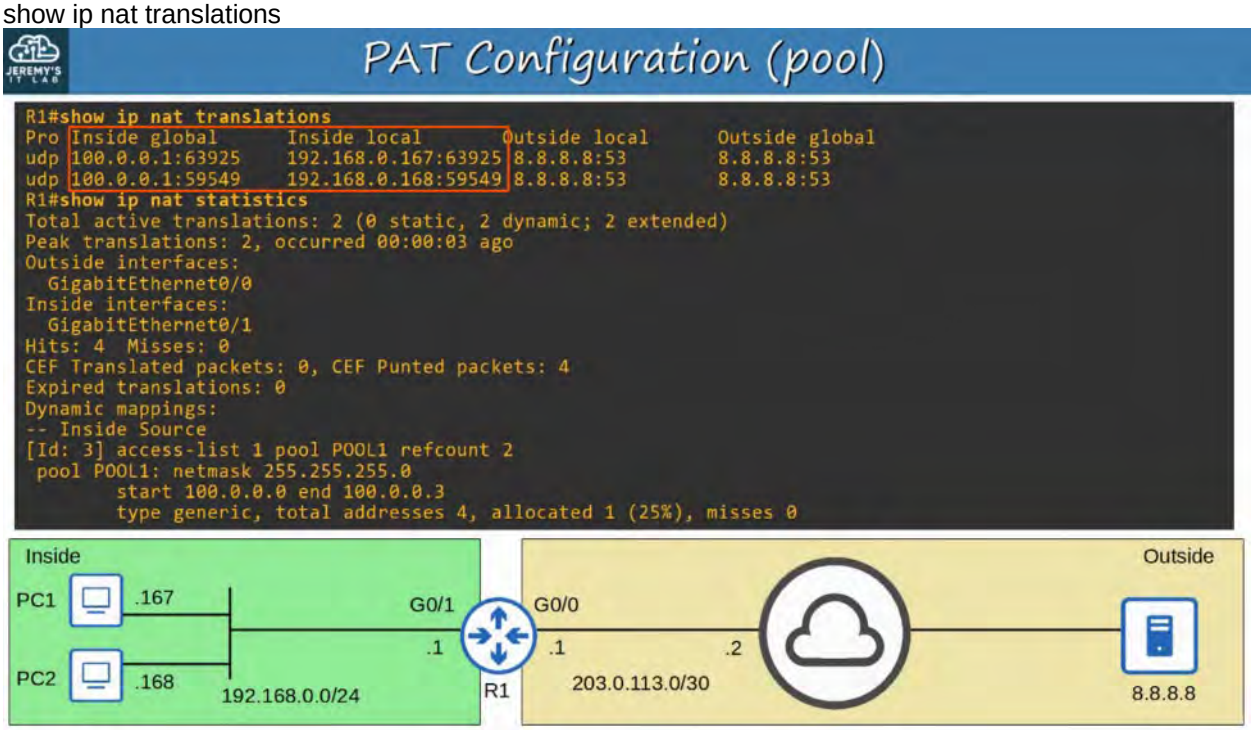
- PAT (NAT OVERLOAD) translates BOTH the IP ADDRESS and the PORT NUMBER (if necessary)
- By using a unique PORT NUMBER for each communication flow, a single PUBLIC IP ADDRESS can be used by many different INTERNAL HOSTS
    - PORT NUMBERS are 16 bits = over 65,000 available port numbers
- The ROUTER will keep track of which INSIDE LOCAL ADDRESS is using which INSIDE GLOBAL ADDRESS and PORT



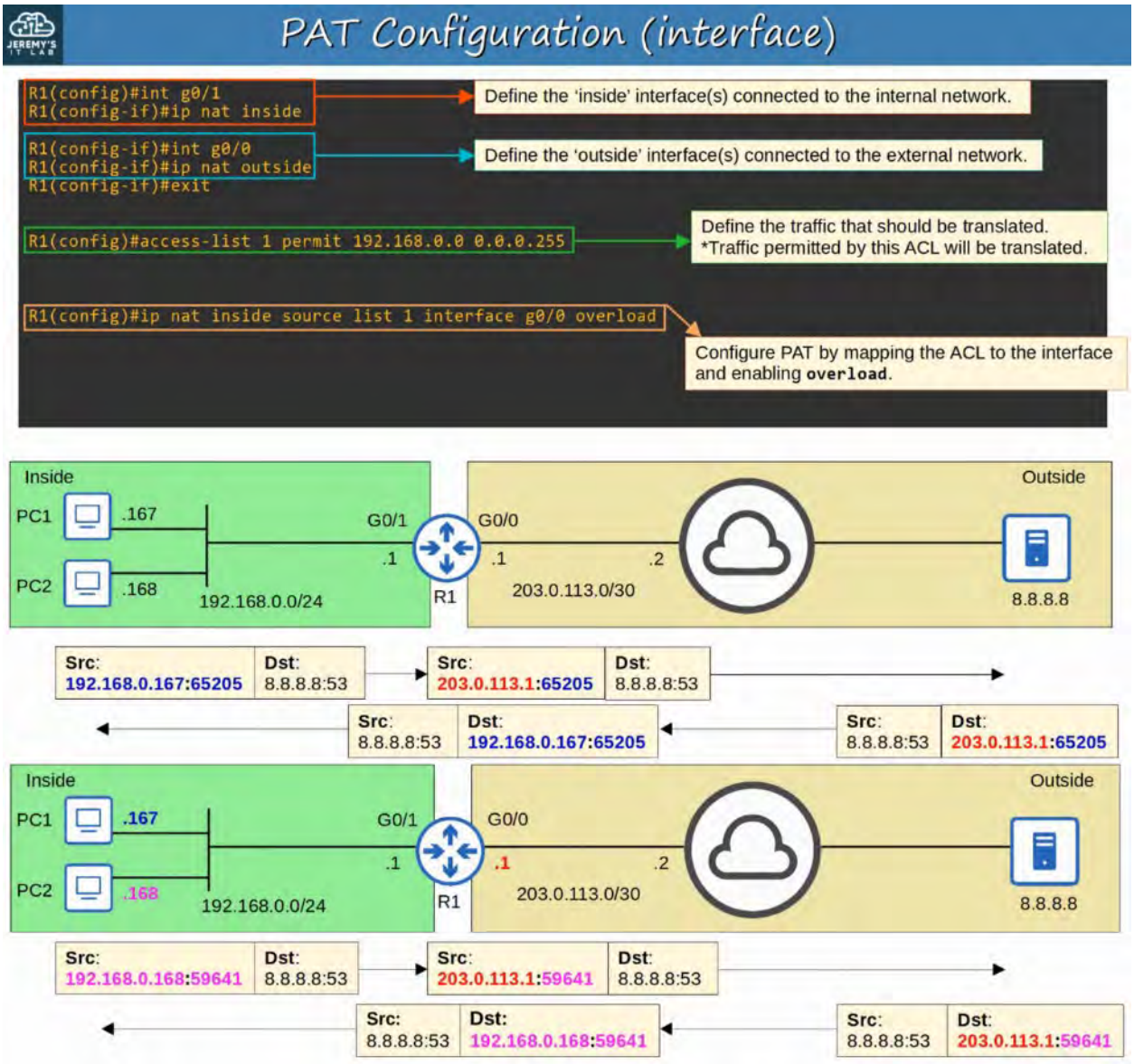
PAT CONFIGURATION (POOL)



show ip nat translations



PAT CONFIGURATION (INTERFACE)



show ip nat translations

```
R1#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
udp 203.0.113.1:65205  192.168.0.167:65205 8.8.8.8:53         8.8.8.8:53
udp 203.0.113.1:59641  192.168.0.168:59641 8.8.8.8:53         8.8.8.8:53
R1#show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:36:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 12  Misses: 0
CEF Translated packets: 0, CEF Punted packets: 12
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 4] access-list 1 interface GigabitEthernet0/0 refcount 2
```

COMMAND REVIEW

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
```

46. QoS (Voice VLANs) : PART 1

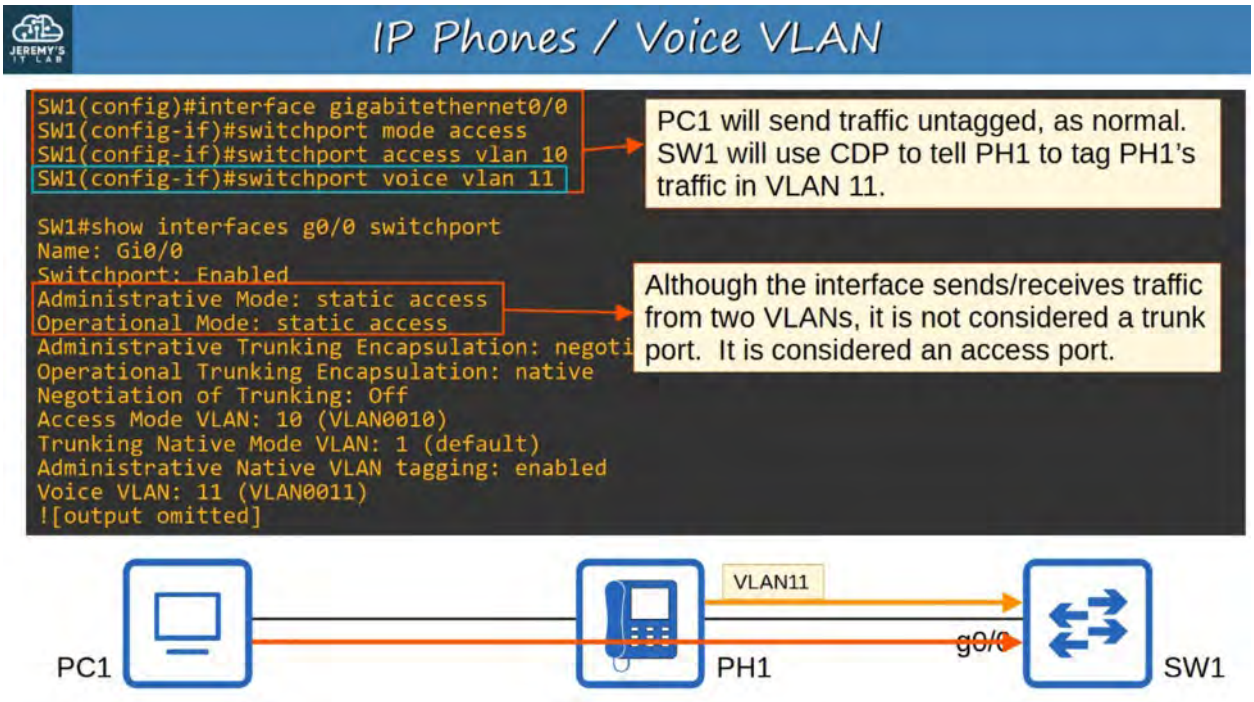
IP PHONES / VOICE LANS

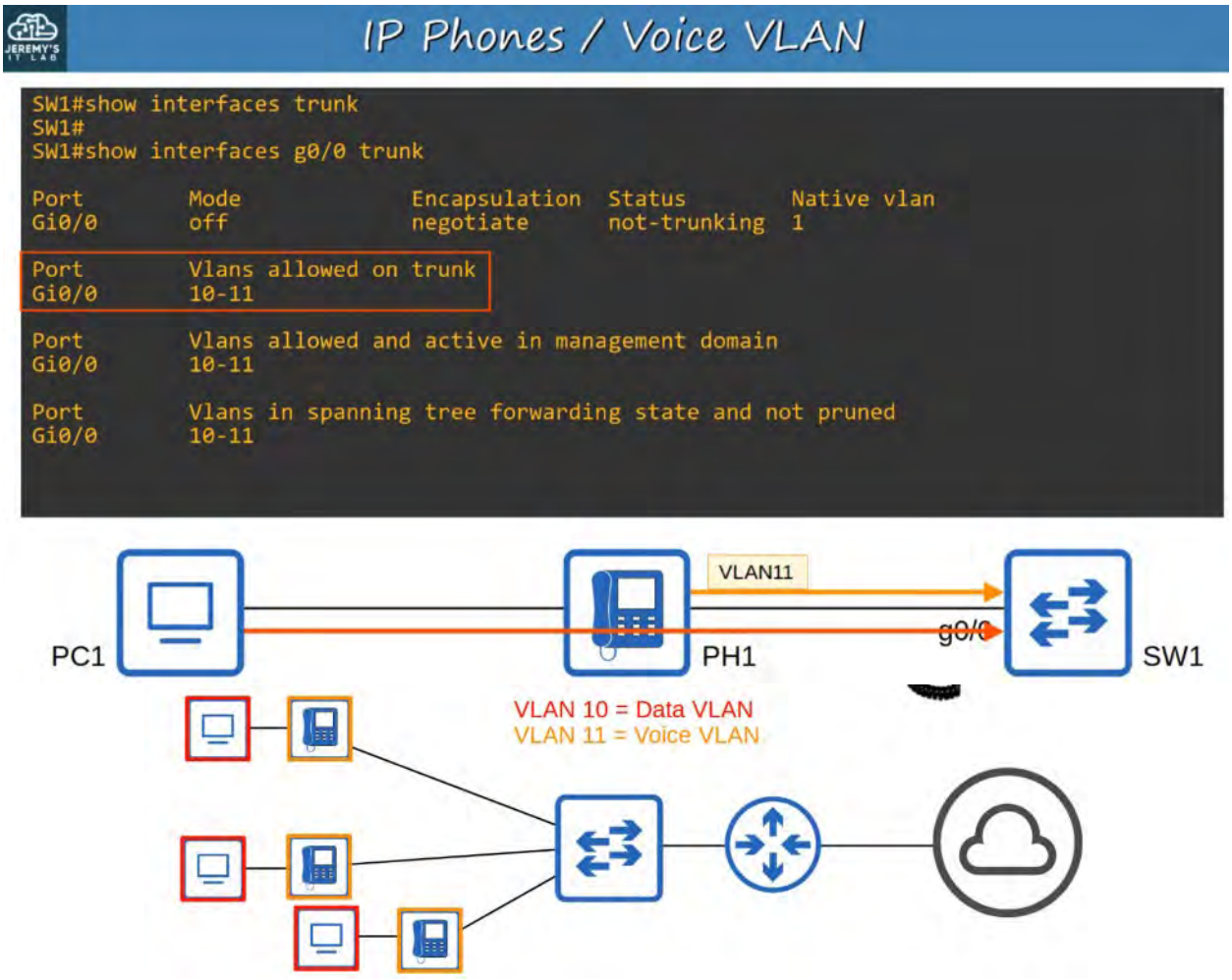
- Traditional phones operate over the *public switched telephone network* (PSTN)
    - Sometimes, this is called POTS (Plain Old Telephone System)
- IP PHONES use VoIP (Voice Over IP) technologies to enable phone calls over an IP NETWORK, such as the INTERNET
- IP PHONES are connected to a SWITCH, just like any other end HOST

IP PHONES

- 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

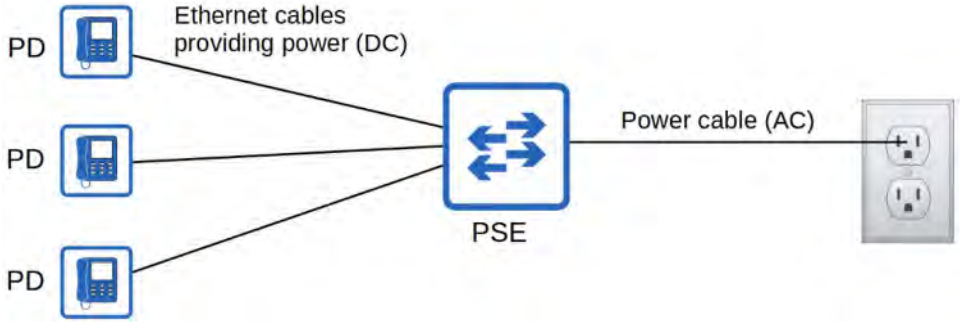
- This allows the PC and the IP PHONE to share a single SWITCH PORT. Traffic from the PC passes through the IP PHONE to the SWITCH
- It is RECOMMENDED to separate “VOICE” traffic (from IP PHONE) and “DATA TRAFFIC” (from the PC) by placing them into SEPARATE VLANS (!)
    - This can be accomplished using 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)

- PoE allows Power Sourcing Equipment (PSE) to provide POWER to Powered Devices (PD) over an ETHERNET cable
- Typically, the PSE is a SWITCH and the PDs are IP PHONES, IP CAMERAS, WIRELESS ACCESS POINTS, etc.
- The PSE receives AC POWER from the outlet, converts it to DC POWER, and supplies that DC POWER to the PDs



- TOO much electrical current can damage electrical DEVICES
- PoE has a process to determine if a CONNECTED DEVICE needs power and how much 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'

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Gi2/1     auto   on         errdisable ok          17.2  16.7
```

    - '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

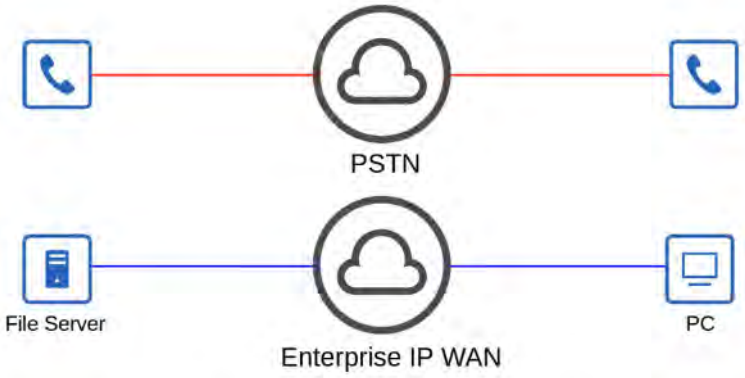
```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police action log
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Gi0/0     auto   on         log        ok          17.2  16.7
```

Power over Ethernet (PoE)

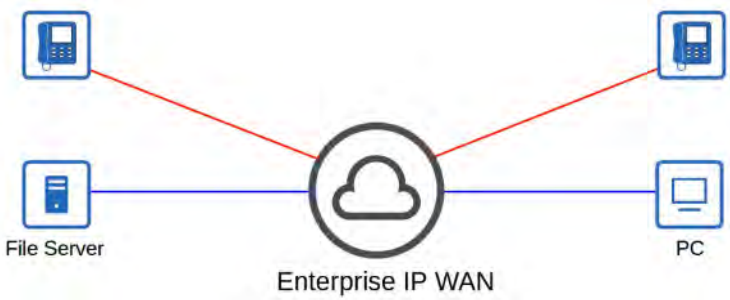
Name	Standard #	Watts	Powered Wire Pairs
Cisco Inline Power (ILP)	Made by Cisco, not standard	7	2
PoE (Type 1)	802.3af	15	2
PoE+ (Type 2)	802.3at	30	2
UPoE (Type 3)	802.3bt	60	4
UPoE+ (Type 4)	802.3bt	100	4

INTRO TO 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



- Modern NETWORKS are typically *converged networks* in which IP PHONES, VIDEO TRAFFIC, REGULAR TRAFFIC, etc. all share the same IP NETWORK
- This enables COST SAVINGS as well as more ADVANCED FEATURES for VOICE and VIDEO TRAFFIC (Example : Collaboration Software like Cisco WebEx, MS 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)

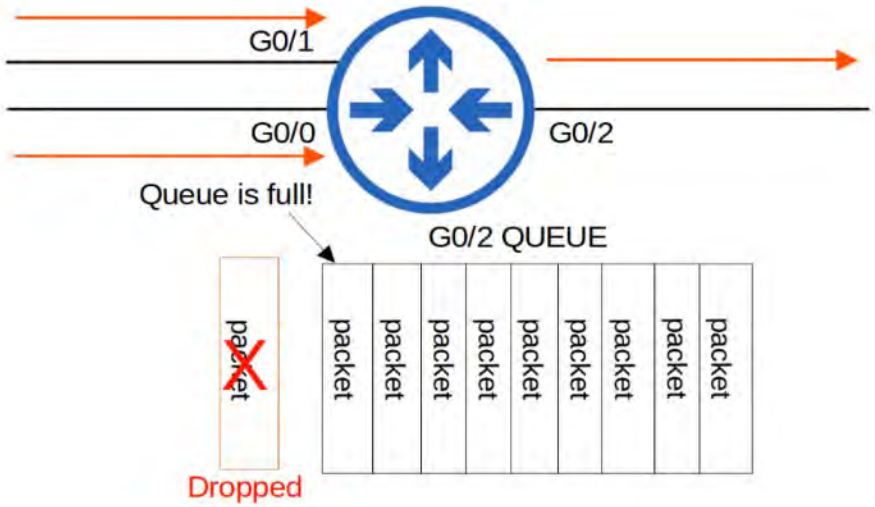
- QoS is used to manage the following characteristics of NETWORK TRAFFIC
  - BANDWIDTH
    - Overall CAPACITY of the LINK (measured in *bits per second*)
    - QoS TOOLS allow you to RESERVE a certain amount of a link's BANDWIDTH for specific kinds of traffic
  - DELAY
    - One-Way Delay = Time it takes traffic to go from SOURCE to DESTINATION
    - Two-Way Delay = Time it takes traffic to go from SOURCE to DESTINATION and return

- 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
- 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

- The FOLLOWING STANDARDS are recommended for ACCEPTABLE INTERACTIVE AUDIO quality:
  - ONE-WAY DELAY : 150 milliseconds or less
  - JITTER : 30 milliseconds or less
  - LOSS : 1% or less
- If these STANDARDS are not met, there could be a noticeable reduction in the QUALITY of the phone call

QoS QUEUING

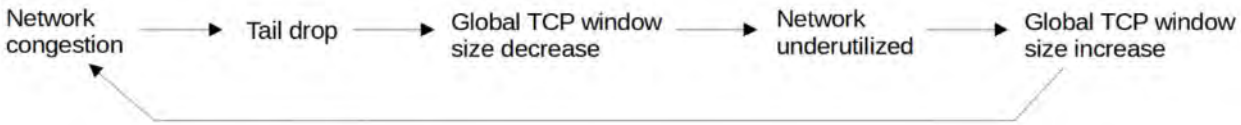
- If a NETWORK DEVICE receives messages FASTER than it can FORWARD them out of the appropriate INTERFACE, the MESSAGES are placed in the QUEUE
- By default, the QUEUED MESSAGES will be FORWARDED in a FIRST IN FIRST OUT (FIFO) manner
  - Message will be SENT in the ORDER they are RECEIVED
- If the QUEUE is FULL, new PACKETS will be DROPPED
- The is called *tail drop*



- 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…



- 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 TCP GLOBAL 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
- WEIGHTED RANDOM EARLY DETECTION (WRED) - an improved version of RED, allows you control which PACKETS are dropped depending on the TRAFFIC CLASS

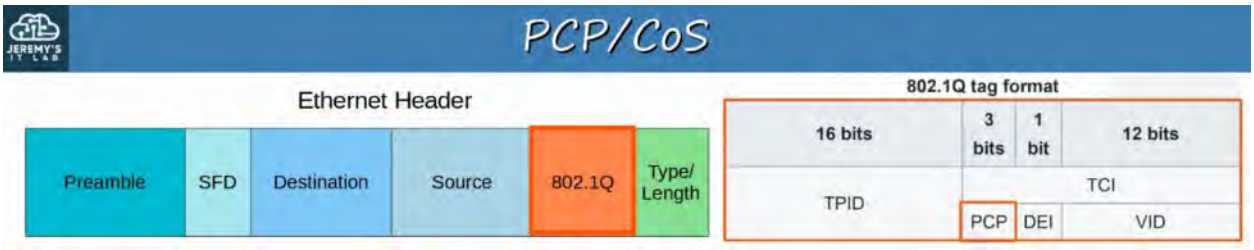
** TRAFFIC CLASSES and details about how QoS works will be covered in DAY 47 **

47. QoS (Quality of Service) : PART 2

CLASSIFICATION / MARKING

- The purpose of QoS is to give certain kinds of NETWORK TRAFFIC priority over other 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
    - 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 kinds 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

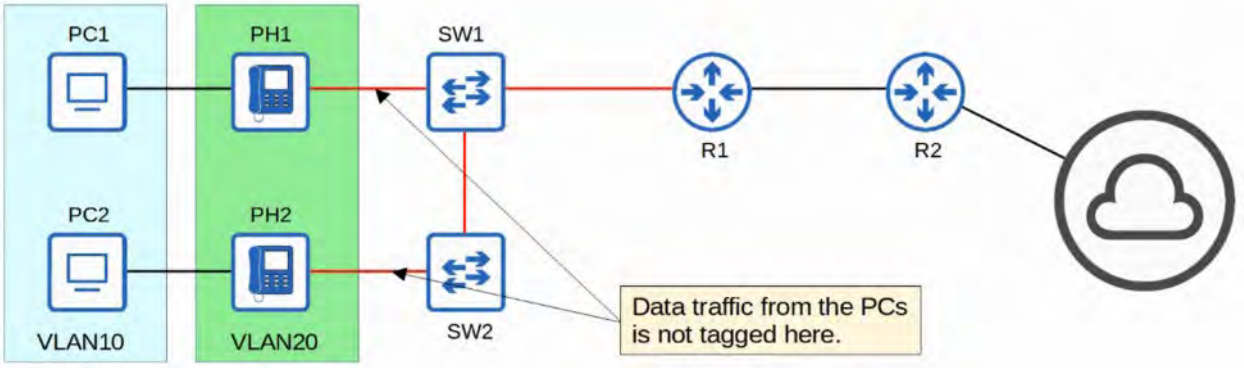


- PCP is also known as CoS (CLASS OF SERVICE)
- It's use is defined by IEEE 802.1p
- 3 bits = 8 possible values (2^3 = 8)

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

- PCP VALUE 0:
    - "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

- PCP VALUE 3 and 5:
    - IP PHONES MARK call signaling TRAFFIC (used to establish calls) as PCP3
        - They MARK the actual VOICE TRAFFIC as PCP5
- Because PCP is found in the dot1q header, it can only be used over the following connections:
    - TRUNK LINKS
    - ACCESS LINKS with a VOICE VLAN
- In the diagram below, TRAFFIC between R1 and R2, or between R2 and EXTERNAL DESTINATIONS will not have a dot1q tag. So, traffic over those links PCP cannot be marked with a PCP value.



THE IP ToS BYTE

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags				Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

ToS byte (current)

DSCP (Differentiated Services Code Point)	ECN

(6 bits for DSCP and 2 bits for ECN)

IP PRECEDENCE (OLD)

IPP (IP Precedence)	(Defined for various purposes, mostly unused)	3 bits = 8 values (0-7)

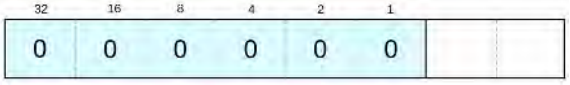
- Standard IPP markings are similar to PCP:
    - 6 and 7 are reserved to ‘network control traffic’ (ie: OSPF Messages between ROUTERS)
    - 5 = VOICE
    - 4 = VIDEO
    - 3 = VOICE SIGNALLING
    - 0 = BEST EFFORT
- With 6 and 7 reserved, 6 possible values remain
- Although 6 values is sufficient for many NETWORKS, the QoS REQUIREMENTS of some NETWORKS demand more flexibility

DSCP (CURRENT)

- 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 on
    - By having generally agreed upon STANDARD MARKINGS for DIFFERENT KINDS of TRAFFIC:
        - QoS DESIGN and IMPLEMENTATION is simplified.
        - QoS works better between ISPs and ENTERPRISES
        - etc.
- 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

DEFAULT FORWARDING (DF)



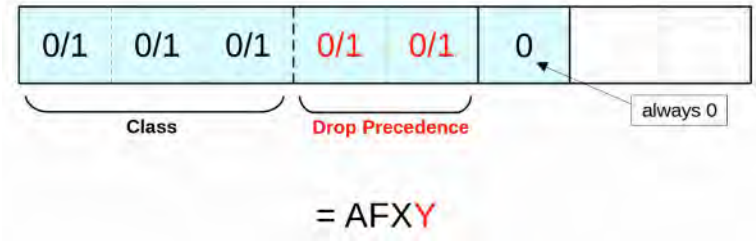
- Used for BEST EFFORT TRAFFIC
- The DSCP marking for DF is 0

EXPEDITED FORWARDING (EF)

- EF is used for TRAFFIC that requires Low Loss / Latency / Jitter
- The DSCP marking for EF is 46

ASSURED FORWARDING (AF)

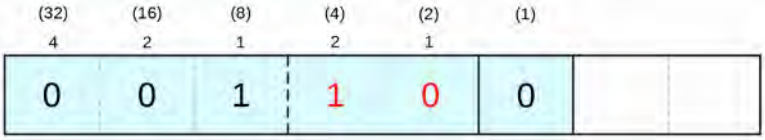
- Defines FOUR TRAFFIC CLASSES
- ALL PACKETS in a CLASS have the same PRIORITY
- Within each CLASS, there are THREE LEVELS of DROP PRECEDENCE
    - HIGHER DROP PRECEDENCE = More likely to DROP the PACKET during CONGESTION



EXAMPLES:

= AF11

(DSCP 10)

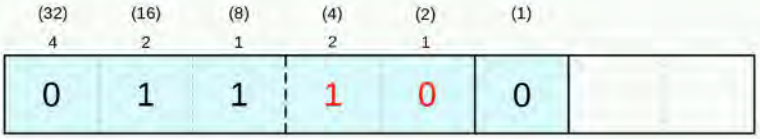


= AF12

(DSCP 12)

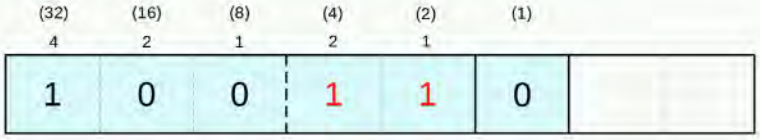
= AF23

(DSCP 22)



= AF32

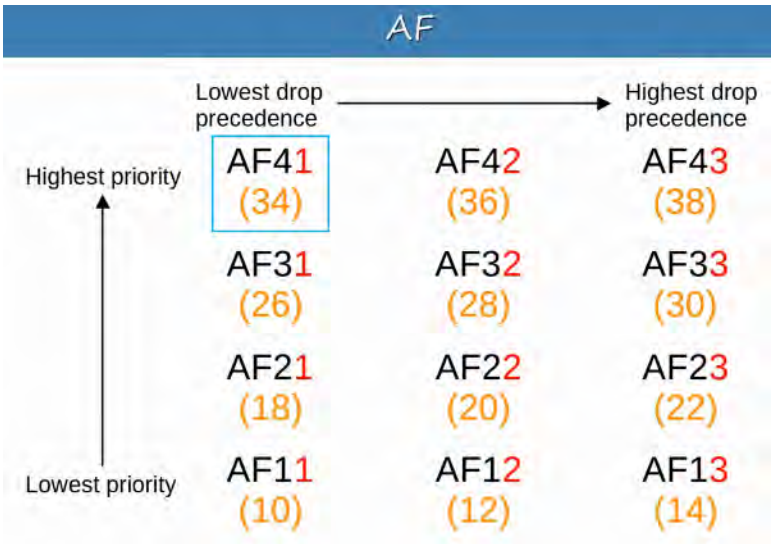
(DSCP 28)



= AF43

(DSCP 38)

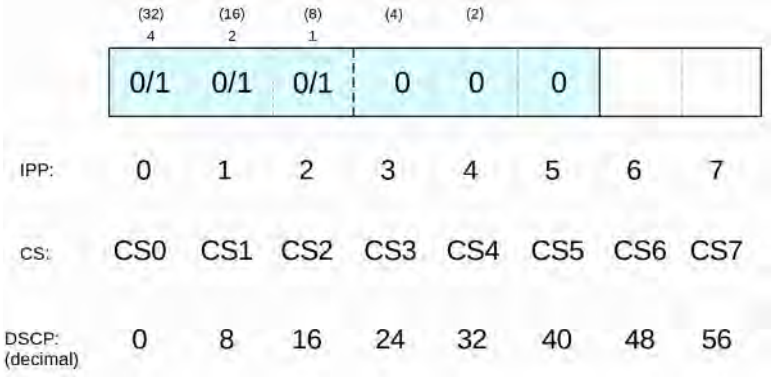
Formula to convert from AF value to decimal DSCP value: $8X + 2Y$



- AF41 gets the BEST TREATMENT (Highest Priority / Lowest Drop)
- AF13 gets the WORST TREATMENT (Lowest Priority / Highest Drop)

CLASS SELECTOR (CS)

- Defines EIGHT DSCP values for backward compatibility with IPP
- The THREE BITS that were added for DSCP are set to 0, and the original IPP bits are used to make 8 values

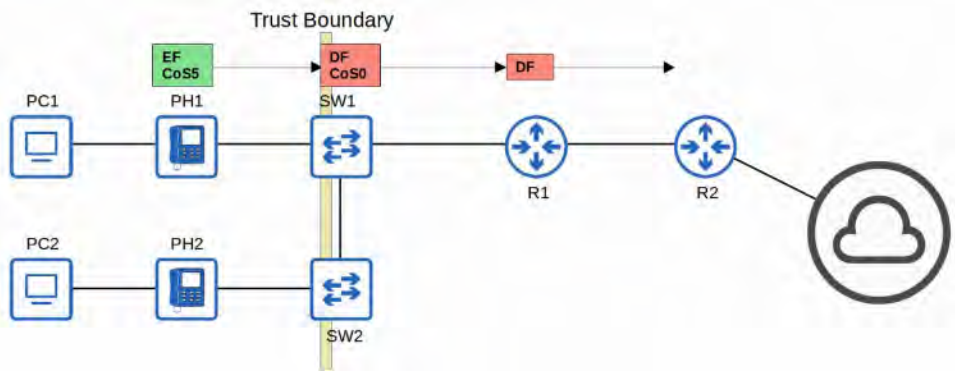


RFC 4954

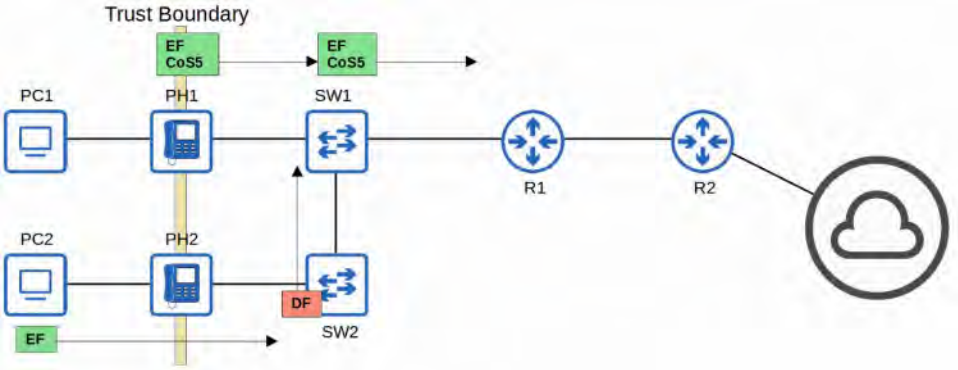
- RFC 4954 was developed with help of Cisco to bring ALL of these VALUES together and STANDARDIZE their use
- The RFC offers MANY specific recommendations, but here are a few KEY ones:
    - VOICE TRAFFIC : EF
    - INTERACTIVE VIDEO : AF4x
    - STREAMING VIDEO : AF3x
    - HIGH PRIORITY DATA : AF2x
    - BEST EFFORT : DF

TRUST BOUNDARIES

- The TRUST BOUNDARY of a NETWORK defines where the DEVICE TRUST / DON'T TRUST the QoS MARKINGS of received messages
- If the MARKINGS are TRUSTED:
    - DEVICE will forward the message without changing the MARKINGS
- If the MARKINGS are NOT TRUSTED:
    - DEVICE will change the MARKINGS according to configured POLICY

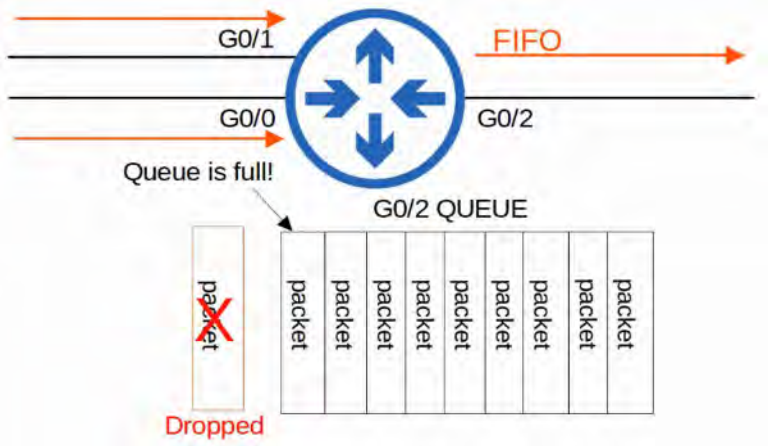


- If an IP PHONE is connected to the SWITCH PORT, it is RECOMMENDED to move the TRUST BOUNDARY to the IP PHONES
- This is done via CONFIGURATION on the SWITCH PORT connected to the IP PHONE
- If a user MARKS their PC's TRAFFIC with a HIGH PRIORITY, the MARKING will be CHANGED (not trusted)



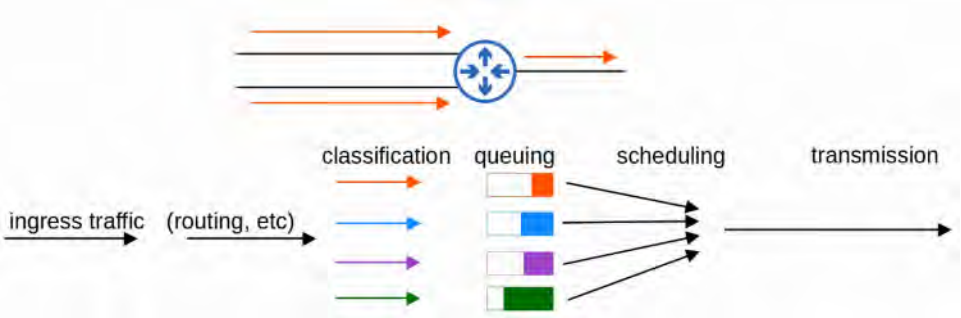
QUEUING / CONGESTION MANAGEMENT

- When a NETWORK DEVICE receives TRAFFIC at a FASTER PACE than it can FORWARD out of the appropriate INTERACE, PACKETS are placed in that INTERFACE'S QUEUE as they wait to be FORWARDED
- When a 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



- An essential part of QoS is the use of MULTIPLE QUEUES
  - This is where CLASSIFICATION plays a role.
  - DEVICE can match TRAFFIC based on various factors (like DSCP MARKINGS in the IP HEADER) and then place it in the appropriate QUEUE

- HOWEVER, the DEVICE is only able to forward one FRAME out of an INTERFACE at once SO a *SCHEDULER*, is used to decide which QUEUE TRAFFIC is FORWARDED from the next
    - *PRIORITZATION* allows the SCHEDULER to give certain QUEUES more PRIORITY than others

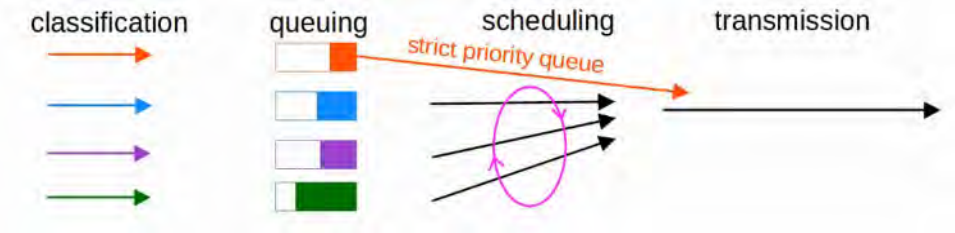


- A COMMON scheduling method is *WEIGHTED ROUND-ROBIN*
    - ROUND-ROBIN:
        - PACKETS taken from each QUEUE in order, cyclically
    - WEIGHTED:
        - More DATA taken from HIGH PRORITY QUEUES each time the SCHEDULER reaches that QUEUE

- CBWFQ (CLASS BASED WEIGHED FAIR QUEUING)
    - Popular method of SCHEDULING
    - Uses WEIGHTED ROUND-ROBIN SCHEDULER while guaranteeing each QUEUE a certain PERCENTAGE of the INTERFACE'S bandwidth during CONGESTION

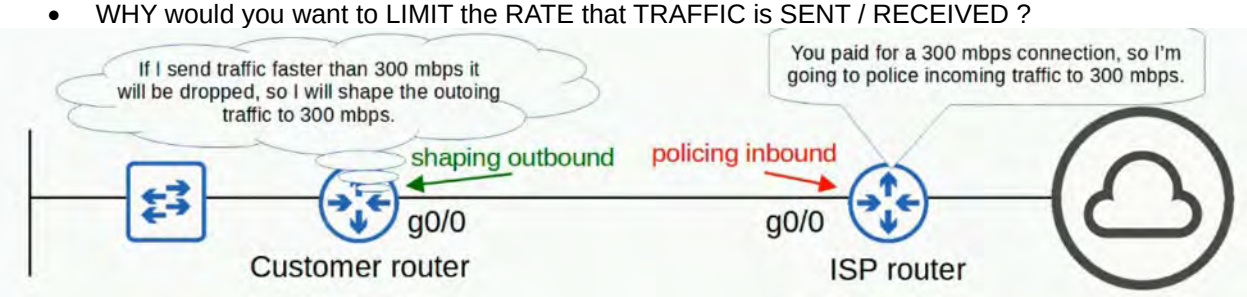
- ROUND-ROBIN SCHEDULING is NOT IDEAL for VOICE / VIDEO TRAFFIC
    - Even if VOICE / VIDEO TRAFFIC receives a guaranteed MINIMUM amount of BANDWIDTH, ROUND-ROBIN can add DELAY and JITTER because even the HIGH PRIORITY QUEUES have to wait their turn in the SCHEDULER

- 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, LLQ has a DOWNSIDE of potentially starving other QUEUES if there is always TRAFFIC in the DESIGNATED *STRICT PRIORITY QUEUE*
        - POLICING 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 / 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
    - POLICING also has the option of RE-MARKING the TRAFFIC, instead of DROPPING
  - “BURST” TRAFFIC over the CONFIGURED RATE is allowed for a short period of time
  - This accommodates DATA APPLICATIONS which typically are “bursty” in nature (ie: not constant stream)
  - The amount of BURST TRAFFIC allowed is configurable
- 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 that TRAFFIC is SENT / RECEIVED ?



48. SECURITY FUNDAMENTALS

KEY SECURITY CONCEPTS
WHY SECURITY?
What is the purpose / goal of SECURITY in an ENTERPRISE ?

- The principles of the CIA TRIAD form the FOUNDATION of SECURITY:
  - CONFIDENTIALITY
    - Only AUTHORIZED USERS should be able to ACCESS DATA
    - Some INFORMATION / DATA is PUBLIC and can be accessed by ANYONE
    - Some INFORMATION / DATA is SECRET and should be only be accessed by SPECIFIC people
  - INTEGRITY
    - DATA should not be tampered with (modified) by unauthorized USERS
    - DATA should be CORRECT and AUTHENTIC
  - AVAILABILITY
    - The NETWORK / SECURITY should be OPERATIONAL and ACCESSIBLE to AUTHORIZED USERS

ATTACKERS can threaten the CONFIDENTIALITY, INTEGRITY, and AVAILBILITY of an enterprise's SYSTEMS and INFORMATION

VULNERABILITY, EXPLOIT, THREAT, MITIGATION

- A VULNERABILITY is any potential weakness that can compromise the CIA of a SYSTEM / INFO
  - A potential weakness isn't a problem in its own
- AN EXPLOIT is something that can potentially be used to exploit the vulnerability
  - Something than can *potentially* be used as an exploit isn't a problem on it's own.
- A THREAT is the potential of a VULNERABILITY to be EXPLOITED
  - A hacker EXPLOITING a VULNERABILITY in your system is a THREAT
- A MITIGATION TECHNIQUE is something that can protect against threats
  - Should be implemented everywhere a VULNERABILITY can be EXPLOITED:
    - Client Devices
    - Servers, Switches, Routers, Firewalls
    - etc.

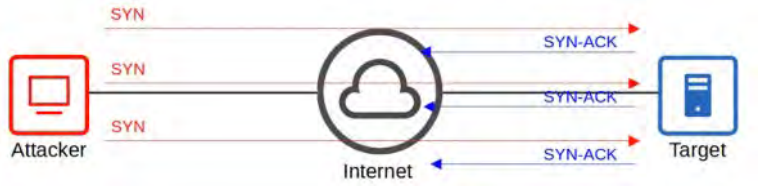
💡 NO SYSTEM IS PERFECTLY SECURE!

COMMON ATTACKS

- DoS (Denial of Service) Attacks
- Spoofing Attacks
- Reflection / Amplification Attacks
- Man-in-the-Middle Attacks
- Reconnaissance Attacks
- Malware
- Social Engineering Attacks
- Password-Related Attacks

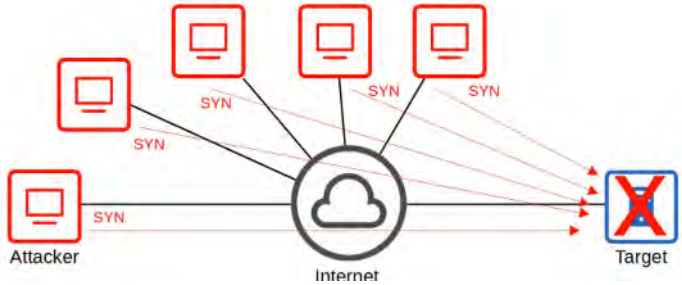
DoS (Denial of Service) Attacks

- DoS attacks threaten the AVAILABILITY of the SYSTEM
- One common DoS attack is the TCP SYN Flood
  - TCP Three-Way Handshake : SYN | SYN-ACK | ACK
  - The ATTACKER sends countless TCP SYN messages to the TARGET
  - The TARGET sends a SYN-ACK message in response to each SYN it receives
  - The ATTACKER never replies with the final ACK of the TCP Three-Way Handshake
  - The incomplete connections fill up the TARGET'S TCP connection table
  - The ATTACKER continues sending SYN messages
  - The TARGET is no longer able to make legitimate TCP connections



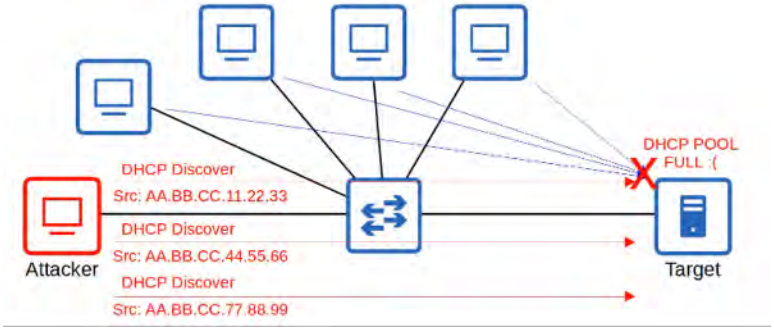
- In a DDoS (Distributed Denial of Service) Attack, the ATTACKER infects many computers with MALWARE and uses them to initiate a Denial-of-Service Attack.
- This group of infected computers is called a BOTNET

Example : A TCP SYN Flood Attack



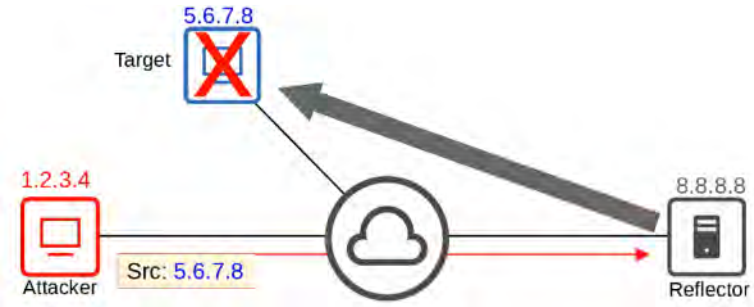
SPOOFING ATTACKS

- To SPOOF an ADDRESS is to use a FAKE SOURCE ADDRESS (IP or MAC)
- Numerous attacks involve spoofing; it's not a SINGLE kind of attack
- An example is a DHCP EXHAUSTION attack
- An ATTACKER uses spoofed MAC ADDRESSES to flood DHCP Discover messages
- The TARGET server's DHCP POOL becomes full, resulting in a Denial-of-Service to other DEVICES



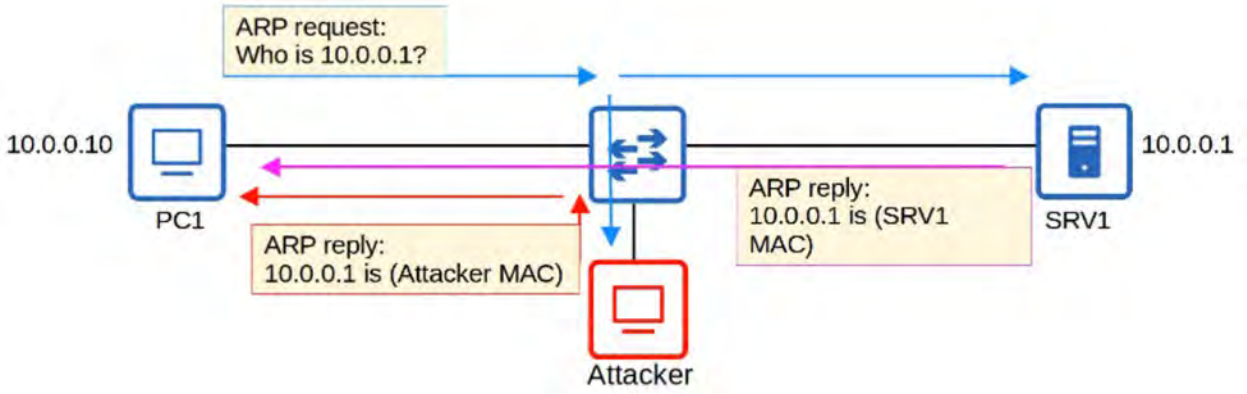
REFLECTION / AMPLIFICATION ATTACKS

- In a REFLECTION attack, the ATTACKER sends traffic to a *reflector*, and spoofs the SOURCE of the PACKET using the TARGET'S IP ADDRESS
- The *reflector* (ie: a DNS Server) sends the reply to the TARGET'S IP ADDRESS
- If the amount of traffic sent to the TARGET is large enough, this can result in a Denial-of-Service
- A REFLECTION attack becomes an AMPLIFICATION attack when the amount of traffic sent by the ATTACKER is small but it triggers a LARGE amount of traffic to be sent from the *reflector* to the TARGET

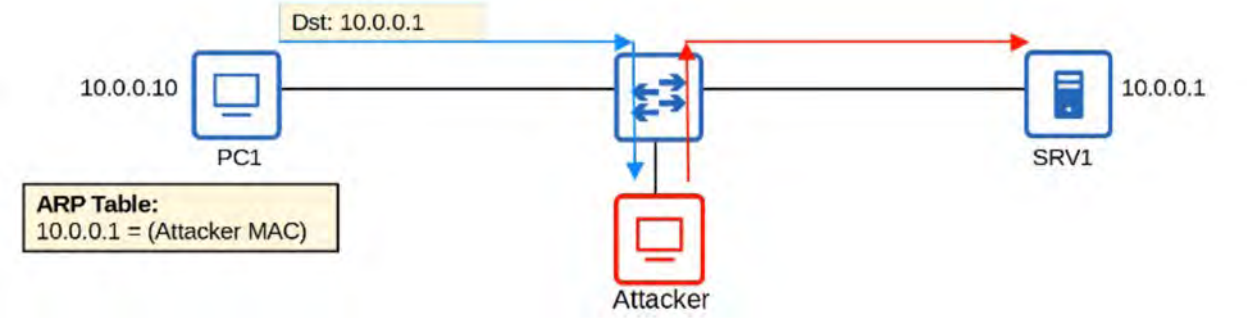


MAN-IN-THE-MIDDLE ATTACKS

- In a MAN-IN-THE-MIDDLE attack, the ATTACKER places himself between the SOURCE and DESTINATION to eavesdrop on communications, or to modify traffic before it reaches the DESTINATION
- A common example is ARP SPOOFING, also known as ARP POISONING
- A HOST sends an ARP REQUEST, asking for the MAC ADDRESS of another DEVICE
- The TARGET of the request sends an ARP REPLY, informing the requester of it's MAC ADDRESS
- The ATTACKER waits and sends another ARP REPLY after it's legitimate replier



- In PC1's ARP table, the entry for 10.0.0.1 will have the ATTACKER'S MAC ADDRESS
- When PC1 tries to send traffic to SRV1, it will be forwarded to the ATTACKER instead
- The ATTACKER can inspect the messages, and then forward them on to SRV1
- The ATTACKER can also modify the messages before forwarding them to SRV1
- This compromises the CONFIDENTIALITY and INTEGRITY of communication between PC1 and SRV1



RECONNAISSANCE ATTACKS

- RECONNAISSANCE ATTACKS are not attacks themselves but they are used to gather information about a TARGET which can be used for a future attack
- This is often publicly available information

- IE: nslookup to learn the IP ADDRESS of a site

- Or a WHOIS query to learn email addresses, phone numbers, physical addresses, etc.

https://lookup.icann.org/lookup

MALWARE

- MALWARE (MALICIOUS SOFTWARE) refers to a variety of harmful programs that can infect a computer
- VIRUSES infect other software (a 'host program')
    - The VIRUS spreads as the software is shared by USERS. Typically, they CORRUPT or MODIFY files on the TARGET computer
- WORMS do not require a host program. They are standalone malware and they are able to spread on their own, without user interaction. They spread of WORMS can congest the NETWORK but the 'payload' of a WORM can cause additional harm to TARGET DEVICES
- TROJAN HORSES are harmful software that is disguised as LEGITIMATE software. They are spread through user interaction such as opening email attachments, downloading a file from the Internet.

The above MALWARE types can exploit various VULNERABILITIES to threaten any of the CIA of a TARGET DEVICE

** There are MANY types of MALWARE

SOCIAL ENGINEERING ATTACKS

- SOCIAL ENGINEERING ATTACKS target the most vulnerable part of ANY system - PEOPLE!
- They involve psychological manipulation to make the TARGET reveal confidential information or perform some action
- PHISHING typically involves fraudulent emails that appear to come from a legitimate business (Amazon, bank, credit card company, etc) and contain links to a fraudulent website that seems legitimate. Users are told to login to the fraudulent website, providing their login credentials to the attacker.
    - SPEAR PHISHING is a more targeted form of phishing, ie: aimed at employees of a certain company
    - WHALING is a phishing targeted at high-profile individuals, ie: a company president
- VISHING (Voice Phishing) is phishing performed over a phone
- SMISHING (SMS Phishing) is phishing using SMS text messages
- WATERING HOLE attacks compromise sites that the TARGET victim frequently visits. If a malicious link is placed on a website the TARGET trusts, they might not hesitate to click it
- TAILGATING attack involves entering restricted, secured areas by simply walking in behind an authorized person as they enter. Often the TARGET will hold the door open for the ATTACKER to be polite, assuming the ATTACKER is also authorized to enter.

PASSWORD-RELATED ATTACKS

- Most systems use a USERNAME / PASSWORD combination to AUTHENTICATE users
- The USERNAME is often simple / easy to guess (for example the user's email address) and the strength and secrecy of the password is relied on to provide the necessary security
- ATTACKERS can learn a user's passwords via multiple methods:
    - Guessing
    - DICTIONARY ATTACK :
        - A program runs through a 'dictionary' or list of common words / passwords to find the TARGET'S password

    - BRUTE FORCE ATTACK :
        - A program tries every possible combination of letters, numbers, and special characters to find the TARGET'S password
- STRONG PASSWORDS should contain:
    - At LEAST 8 characters (preferably more)
    - A mixture of UPPERCASE and LOWERCASE letters
    - A mixture of LETTERS and NUMBERS
    - One of more SPECIAL CHARACTERS (# @ ! ? etc.)
    - Should be CHANGED REGULARLY

PASSWORDS / MULTI-FACTOR AUTHENTICATION (MFA)

- MULTI-FACTOR AUTHENTICATION involves providing more than just a USERNAME / PASSWORD to prove your identity
- It usually involves providing TWO of the following (= Two-Factor Authentication) :
    - SOMETHING YOU KNOW
        - A USERNAME / PASSWORD combination, a PIN, etc.
    - SOMETHING YOU HAVE
        - Pressing a notification that appears on your phone, a badge that is scanned, etc.
    - SOMETHING YOU ARE
        - Biometrics such as a face scan, palm scan, fingerprint scan, retina scan, etc.
- Requiring multiple factors of AUTHENTICATION greatly increases the security. Even if the ATTACKER learns the TARGET'S PASSWORD (SOMETHING YOU KNOW), they won't be able to login to the TARGET'S account

---

DIGITAL CERTIFICATES

- DIGITAL CERTIFICATES are another form of AUTHENTICATION used to prove the identity of the holder of the certificate
- They are used for websites to verify that the website being accessed is legitimate
- Entities that want a certificate to prove their identity send a CSR (CERTIFICATE SIGNING REQUEST) to a CA (CERTIFICATE AUTHORITY) which will generate and sign the certificate

---

CONTROLLING AND MONITORING USERS WITH AAA

- AAA (Triple-A) stands for AUTHENTICATION, AUTHORIZATION, and ACCOUNTING
- It is a framework for controlling and monitor users of a computer system (ie: a network)
- AUTHENTICATION
    - Process of verifying a user's identity
    - Logging in = AUTHENTICATION
- AUTHORIZATION
    - Process of granting the user the appropriate access and permissions
    - Granting the user access to some files / services, restricting access to other files / services = AUTHORIZATION
- ACCOUNTING
    - Process of recording the user's activities on the system
    - Logging when a user makes a change to a file = ACCOUNTING
- Enterprises typically use a AAA server to provide AAA services
    - ISE (Identity Services Engine) is Cisco's AAA server
- AAA Servers usually support the following TWO AAA Protocols:
    - RADIUS : Open Standard Protocol
        - Uses UDP PORTS 1812 and 1813
    - TACACS+ : Cisco Proprietary Protocol
        - Uses TCP PORT 49

💡 FOR THE CCNA, KNOW THE DIFFERENCES BETWEEN AUTHENTICATION, AUTHORIZATION, and ACCOUNTING

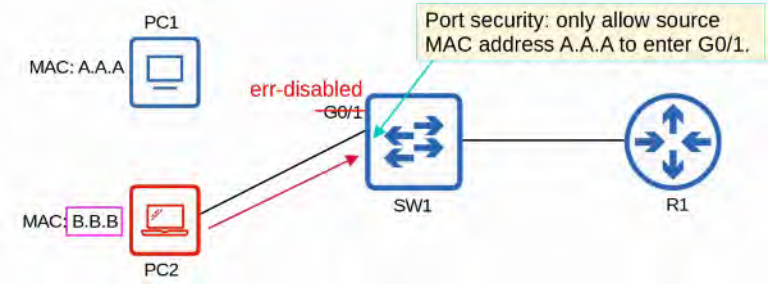
SECURITY PROGRAM ELEMENTS

- USER AWARENESS PROGRAMS are designed to make employees aware of potential security threats and risks
- USER TRAINING PROGRAMS are formal than USER AWARENESS PROGRAMS
- PHYSICAL ACCESS CONTROL protect equipment and data from potential attackers by only allowing authorized users into the protected areas such as NETWORK CLOSETS or DATA CENTER FLOORS

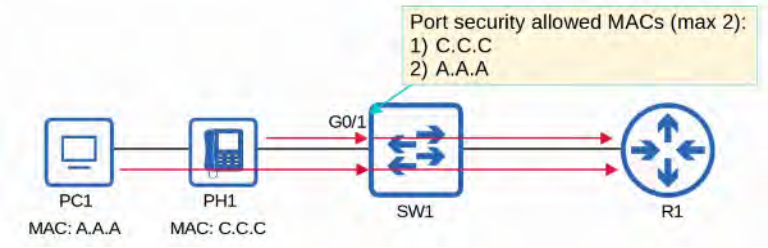
49. PORT SECURITY

INTRO TO 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



- 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 DO NOT 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
- A COMBINATION of manually configured MAC ADDRESSES and DYNAMICALLY LEARNED ADDRESSES is possible



WHY USE 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 is 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 (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

ENABLING PORT SECURITY

```
SW1(config)#interface g0/1
SW1(config-if)#switchport port-security
Command rejected: GigabitEthernet0/1 is a dynamic port.

SW1(config-if)#do show int g0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
![output omitted]

SW1(config-if)#switchport mode access

SW1(config-if)#do show int g0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access

SW1(config-if)#switchport port-security
SW1(config-if)#
```

Port security can be enabled on access ports or trunks ports, but they must be statically configured as access or trunk.

`switchport mode access` = OK

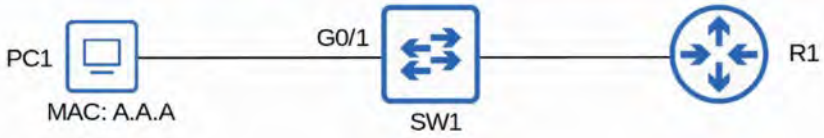
`switchport mode trunk` = OK

~~`switchport mode dynamic auto`~~

~~`switchport mode dynamic desirable`~~

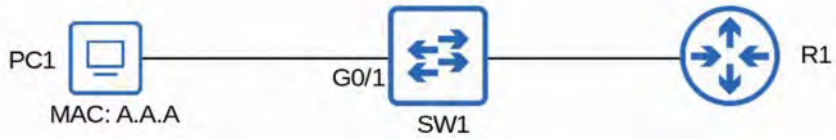
The administrative mode is now static access, so the **`switchport port-security`** command should work.

The command works, so port security is now enabled on G0/1.

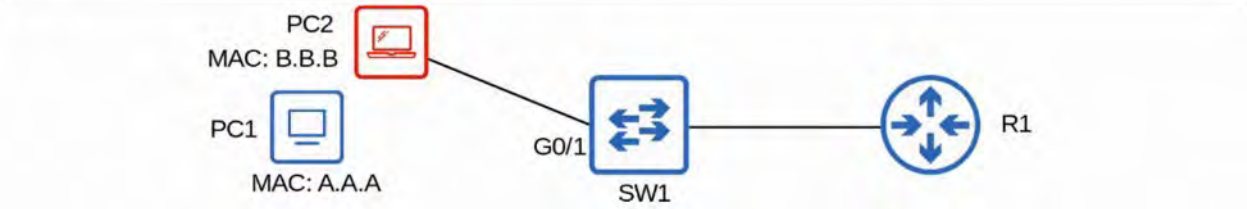


show port-security interface

```
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
```



```
SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000b.000b.000b:1
Security Violation Count   : 1
```



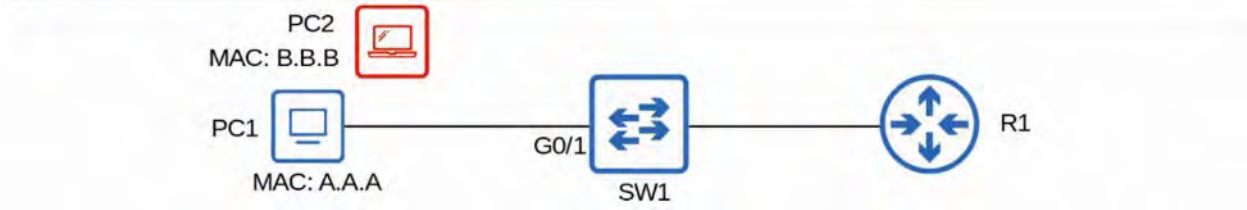

```
SW1#show interfaces status

Port      Name               Status       Vlan       Duplex  Speed Type
Gi0/0                        connected    1            auto   auto unknown
Gi0/1                        err-disabled 1            auto   auto unknown
```

RE-ENABLING AN INTERFACE (MANUALLY)

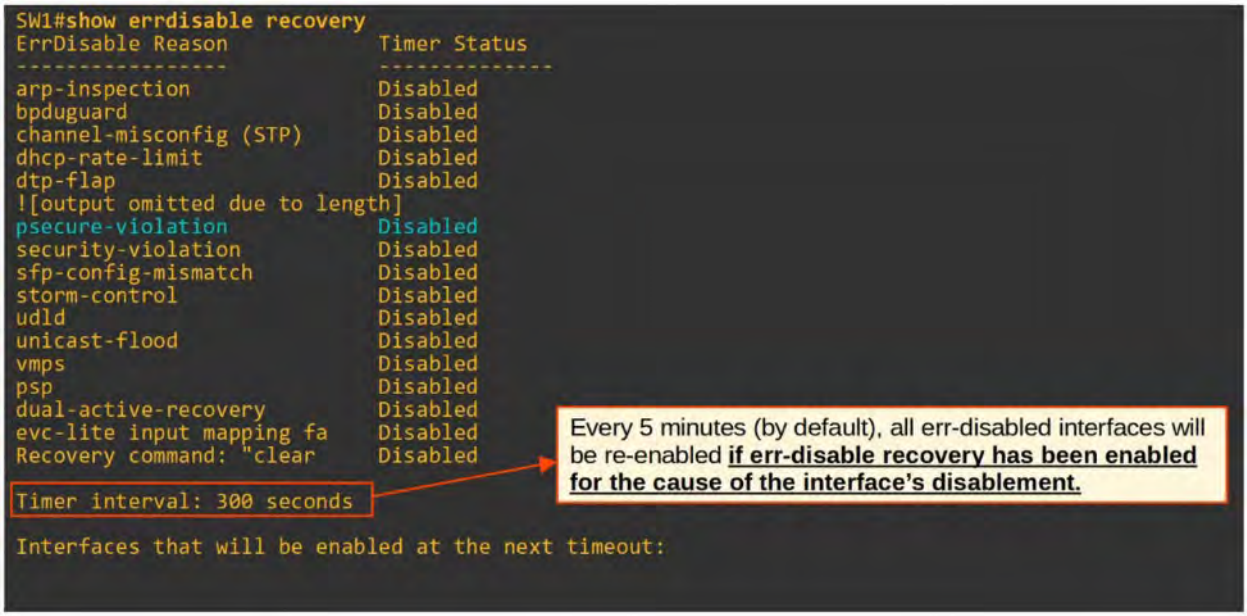
```
SW1(config)#interface g0/1
SW1(config-if)#shutdown
SW1(config-if)#no shutdown
```

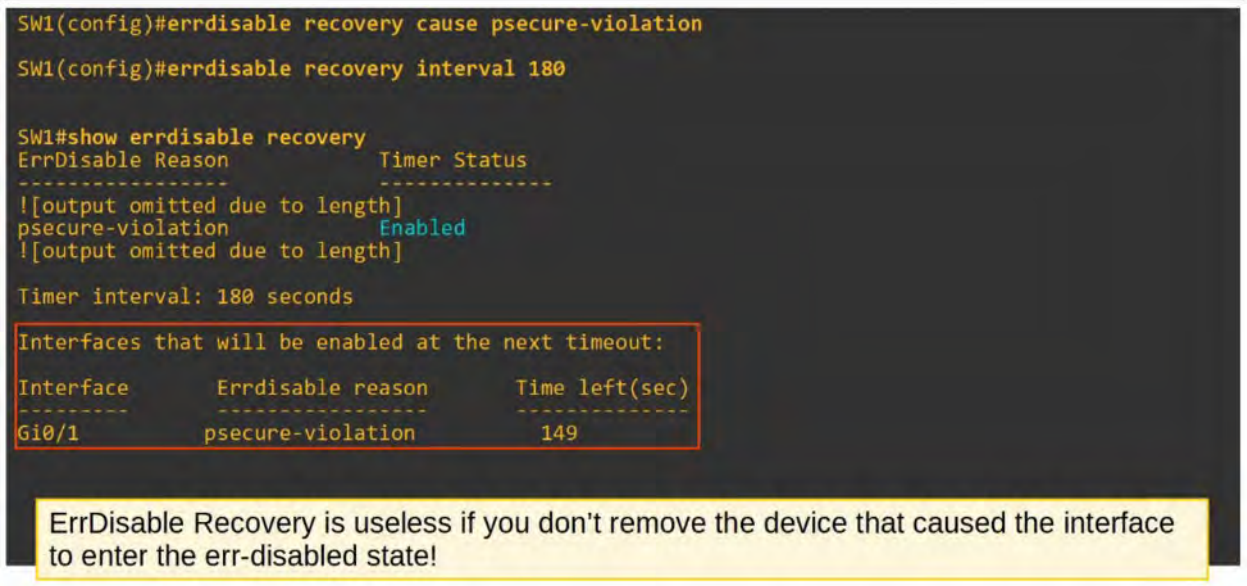
1) Disconnect the unauthorized device
2) **shutdown** and then **no shutdown** the interface

```
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        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
```



RE-ENABLING AN INTERFACE (ERR-DISABLE RECOVERY)





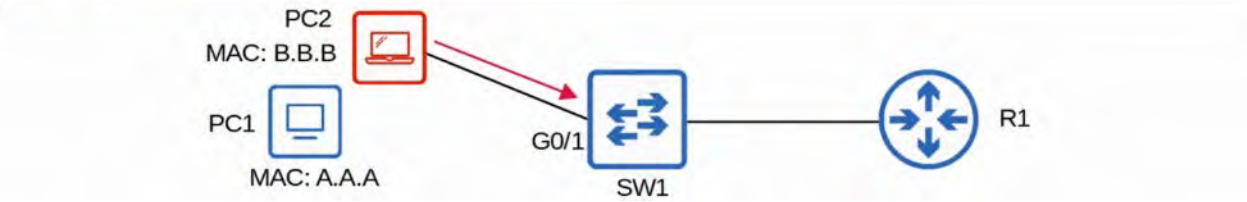
VIOLATION MODES

- 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 a SYSLOG / SNMP message for unauthorized traffic
- It does NOT increment the VIOLATION counter

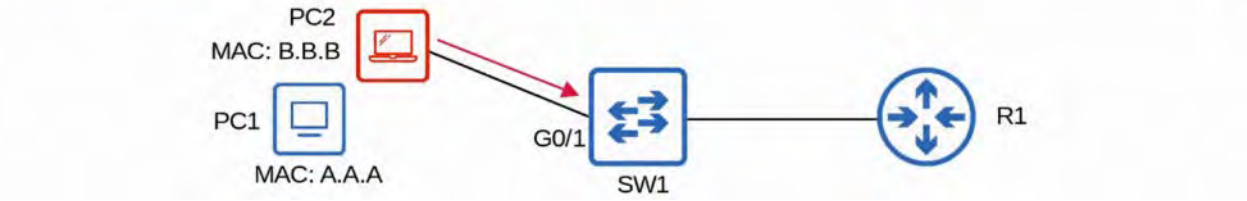
VIOLATION MODE - RESTRICT

```
SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address 000a.000a.000a
SW1(config-if)#switchport port-security violation restrict

*May 23 22:54:09.951: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC
address 000b.000b.000b on port GigabitEthernet0/1.

SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Restrict
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000b.000b.000b:1
Security Violation Count   : 12
```



VIOLATION MODE - PROTECT

```
SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address 000a.000a.000a
SW1(config-if)#switchport port-security violation protect


SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Protect
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000b.000b.000b:1
Security Violation Count   : 0
```



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 (address configured with switchport port-security mac-address x.x.x) is DISABLED by DEFAULT

```
SW1(config-if)#switchport port-security aging time 30
SW1(config-if)#switchport port-security aging type inactivity
SW1(config-if)#switchport port-security aging static


SW1#show port-security interface g0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 30 mins
Aging Type                 : Inactivity
SecureStatic Address Aging : Enabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 1
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 000a.000a.000a:1
Security Violation Count   : 0

SW1#show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
               (Count)       (Count)          (Count)
-----------------------------------------------------------------------------
      Gi0/1              1            1                  0         Shutdown
-----------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 4096
```

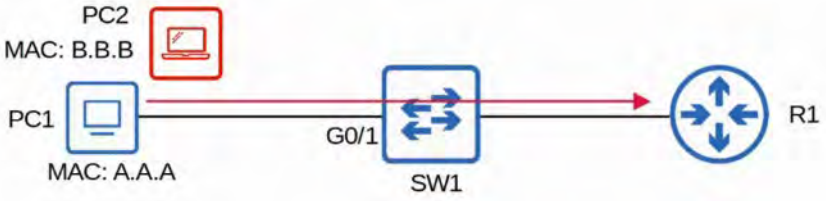
STICKY SECURE MAC ADDRESSES

- 'STICKY' SECURE MAC ADDRESS learning can be enabled with the following command:
    - SW(config-if)# switchport port-security mac-address sticky
- When enabled, dynamically-learned SECURE MAC ADDRESSES will be added to the running configuration, 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 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

Sticky Secure MAC Addresses

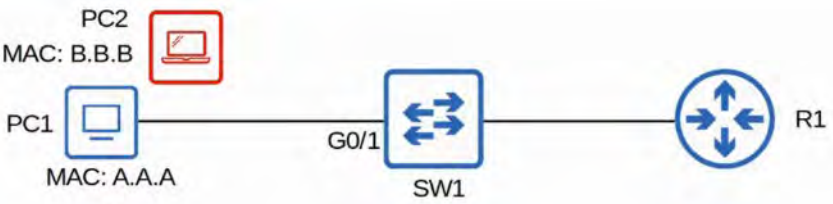
```
SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address sticky
SW1(config-if)#do show running-config interface g0/1
!
interface GigabitEthernet0/1
 switchport mode access
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 000a.000a.000a
 switchport port-security
 negotiation auto
```



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 ADDRESSES 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
```



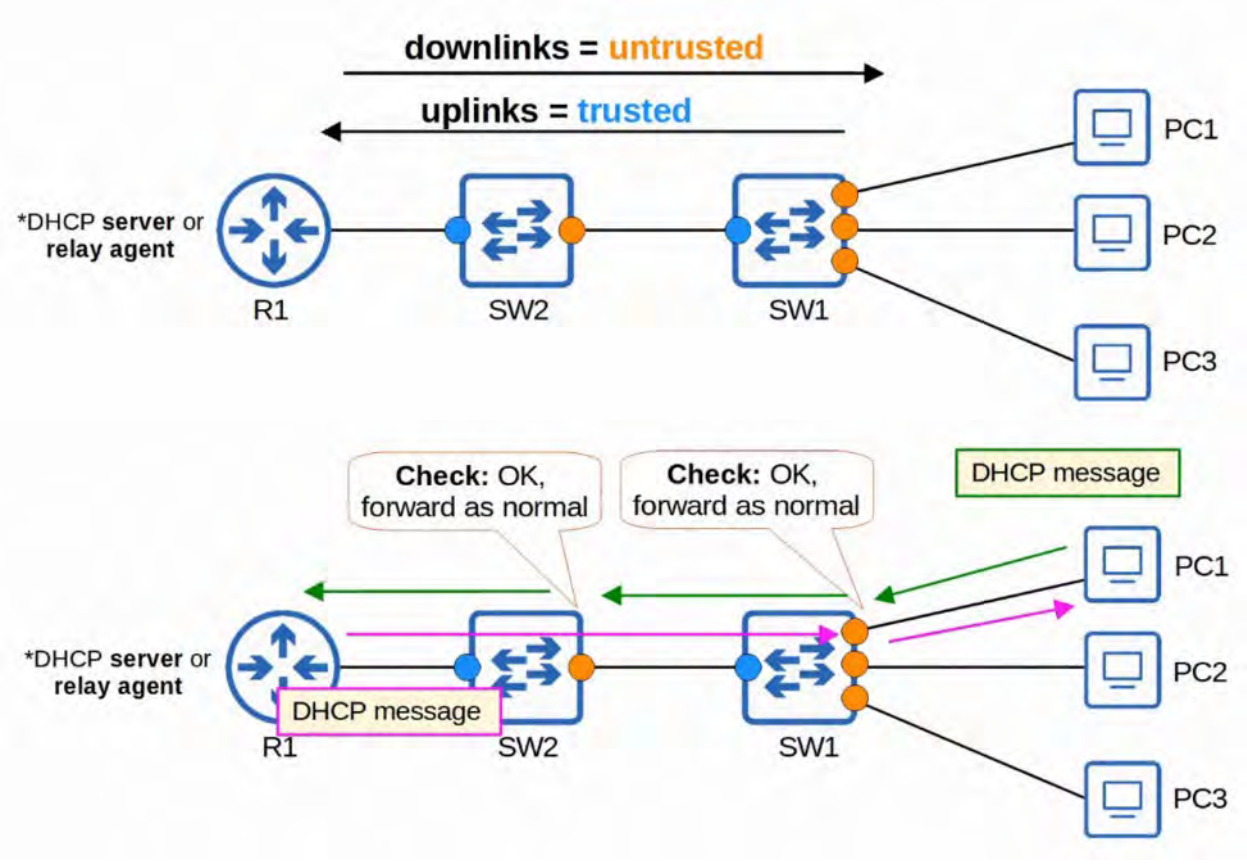
COMMAND REVIEW

```
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
```

50. DHCP SNOOPING (LAYER 2)

WHAT IS DHCP SNOOPING?

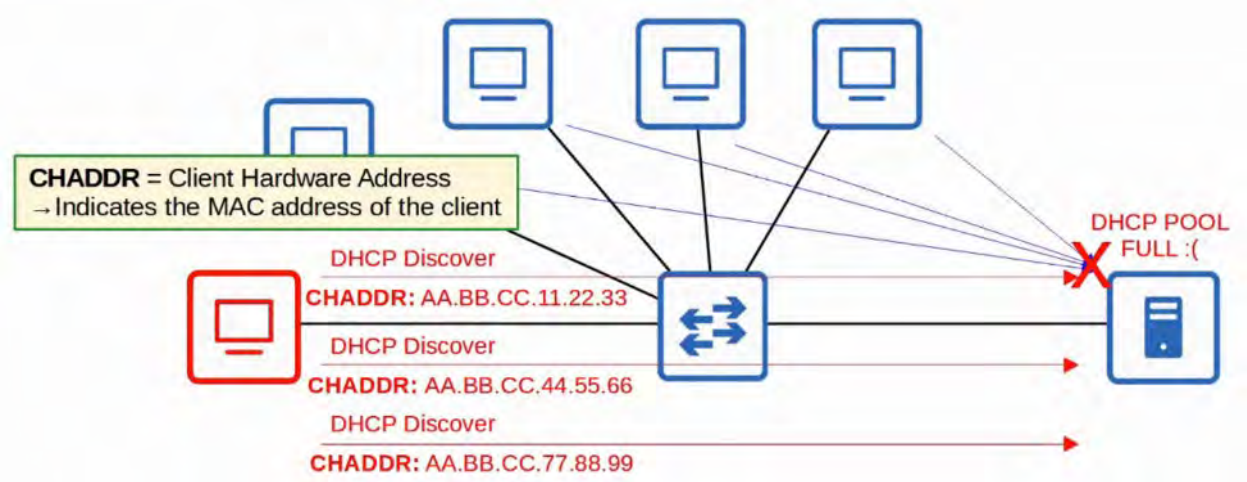
- DHCP SNOOPING is a security feature of SWITCHES that is used to filter DHCP messages received on UNTRUSTED PORTS
- DHCP SNOOPING only filters DHCP MESSAGES.
    - Non-DHCP MESSAGES are not affected
- All PORTS are UNTRUSTED, by DEFAULT
    - Usually UPLINK PORTS are configured as TRUSTED PORTS, and DOWNLINK PORTS remain UNTRUSTED



ATTACKS ON DHCP

DHCP STARVATION

- An example of a DHCP-based ATTACK is a DHCP STARVATION ATTACK
- An ATTACKER uses spoofed MAC ADDRESSES to flood DHCP DISCOVER messages
- The TARGET server's DHCP POOL becomes full, resulting in a DoS to other DEVICES

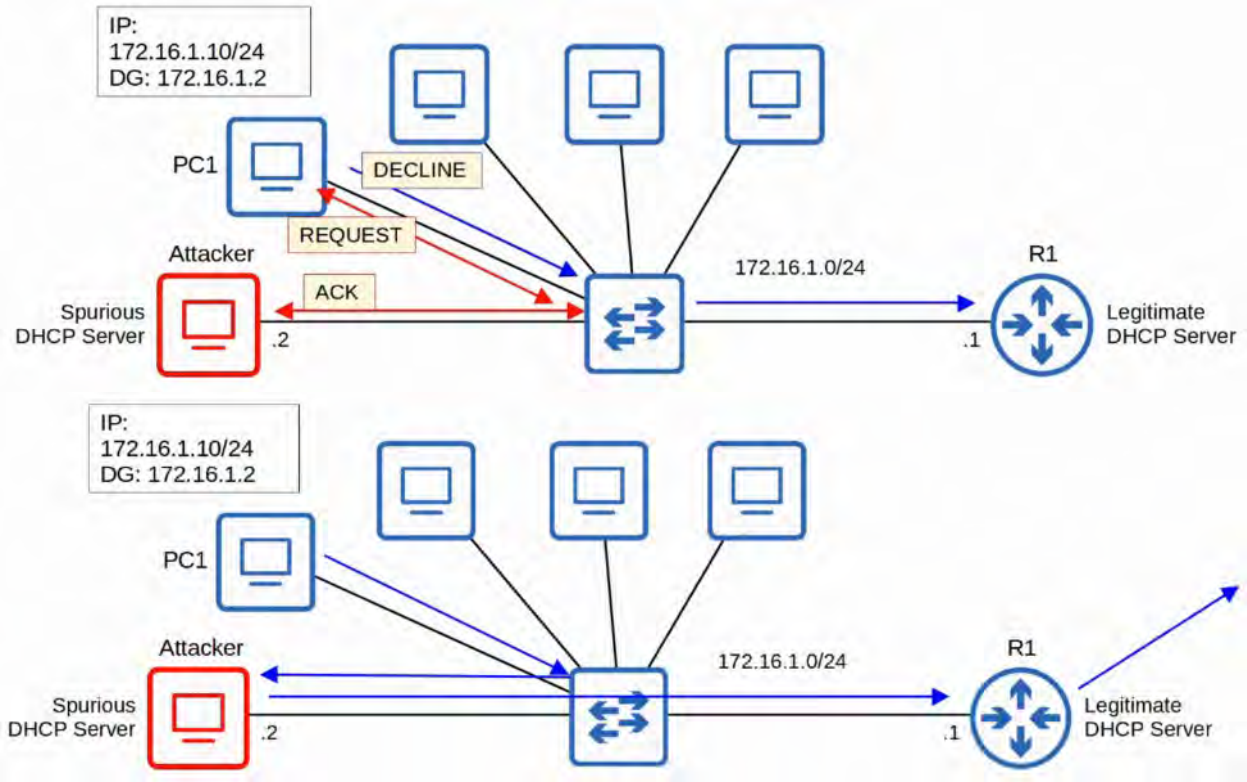


DHCP POISONING (Man-in-the-Middle)

- 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 CLIENTS use the *spurious SERVER'S IP* as a DEFAULT GATEWAY

** CLIENTS usually accept the first DHCP 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



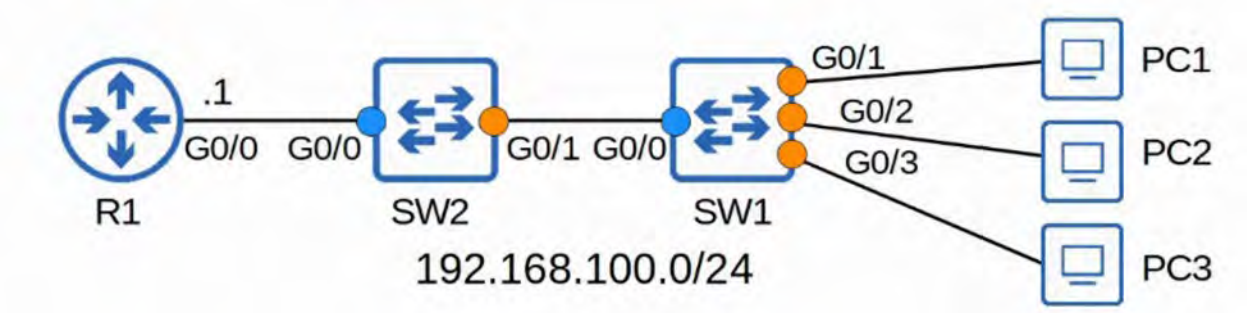
DHCP MESSAGES

- 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

HOW DOES IT WORK?

- 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 as 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 SNOOPING CONFIGURATION



SWITCH 2's CONFIGURATION

```
SW2(config)#ip dhcp snooping
SW2(config)#ip dhcp snooping vlan 1
SW2(config)#no ip dhcp snooping information option
SW2(config)#interface g0/0
SW2(config-if)#ip dhcp snooping trust
```

I will explain this later!

SWITCH 1's CONFIGURATION

```
SW1(config)#ip dhcp snooping
SW1(config)#ip dhcp snooping vlan 1
SW1(config)#no ip dhcp snooping information option
SW1(config)#interface g0/0
SW1(config-if)#ip dhcp snooping trust

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
```

RELEASE/DECLINE messages will be checked to make sure their IP address/interface ID match the entry in the DHCP snooping table.

DHCP SNOOPING RATE-LIMITING

- DHCP SNOOPING can limit the RATE at which DHCP messages are allowed to enter an INTERFACE
- If the RATE of DHCP messages crosses the configured LIMIT, the INTERFACE is err-disabled
- Like with PORT SECURITY, the interface can be manually re-enabled, or automatically re-enabled with errdisable recovery

```
SW1(config)#interface range g0/1 - 3
SW1(config-if-range)#ip dhcp snooping limit rate 1

*Jun  5 13:15:14.180: %DHCP_SNOOPING-4-DHCP_SNOOPING_ERRDISABLE_WARNING: DHCP Snooping received 1 DHCP packets on
interface Gi0/1
*Jun  5 13:15:14.181: %DHCP_SNOOPING-4-DHCP_SNOOPING_RATE_LIMIT_EXCEEDED: The interface Gi0/1 is receiving more
than the threshold set
*Jun  5 13:15:14.182: %PM-4-ERR_DISABLE: dhcp-rate-limit error detected on Gi0/1, putting Gi0/1 in err-disable
state
*Jun  5 13:15:15.185: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Jun  5 13:15:16.190: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to down
```

- You wouldn't set the limit rate to 1 since it's so low, it would shut the port immediately but this shows how RATE-LIMITING works

errdisable recovery cause dhcp-rate-limit

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

SW1#show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
arp-inspection               Disabled
bpduguard                    Disabled
channel-misconfig (STP)      Disabled
dhcp-rate-limit              Enabled
dtp-flap                     Disabled
gbic-invalid                 Disabled
inline-power                 Disabled
![output omitted due to length]


Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

Interface       Errdisable reason       Time left(sec)
---------       -----------------       --------------
Gi0/1            dhcp-rate-limit          293
```

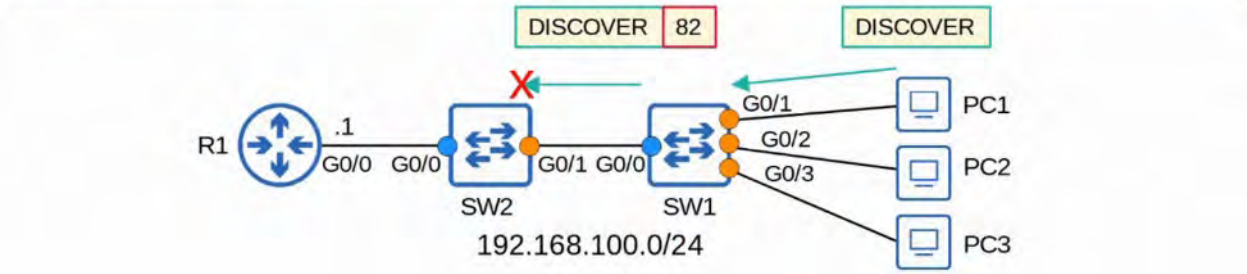
Rate-limiting can be very useful to protect against DHCP exhaustion attacks.

DHCP OPTION 82 (INFORMATION OPTION)

- OPTION 82, also known as a 'DHCP RELAY AGENT INFOMRATION 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 message 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

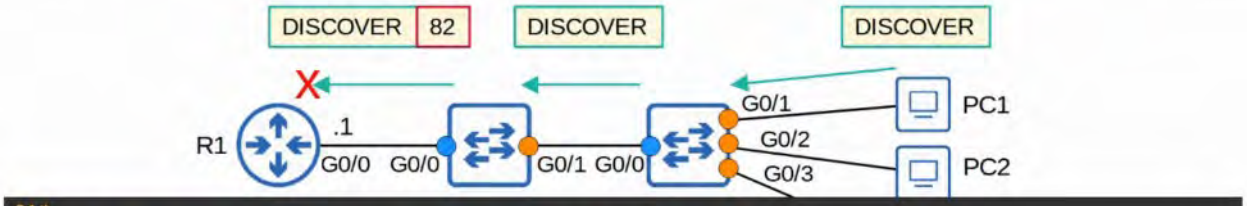
```
SW2#
*Jun  6 01:36:15.298: %DHCP_SNOOPING-5-DHCP_SNOOPING_NONZERO_GIADDR: DHCP_SNOOPING drop message with non-
zero giaddr or option82 value on untrusted port, message type: DHCPDISCOVER, MAC sa: 0c29.2f67.e900
```



THIS command disables OPTION 82 for SW1 but NOT SW2

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

TRAFFIC gets passed to R1 and is DROPPED because of “inconsistent relay information” (packet contains OPTION 82 but wasn’t dropped by SW2)



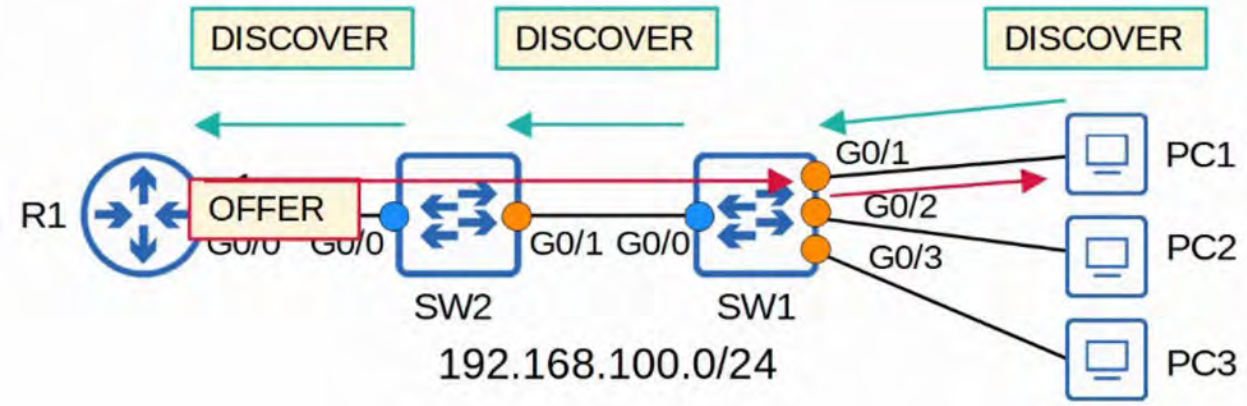

```
R1#
*Jun  6 01:46:46.763: DHCPD: inconsistent relay information.
*Jun  6 01:46:46.763: DHCPD: relay information option exists, but giaddr is zero.
```

By ENABLING OPTION 82 on both SWITCHES…

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

```
SW2(config)#no ip dhcp snooping information option
```

PC1’s DHCP DISCOVER message gets passed, through SW1 and SW2, to R1. R1 responds with an DHCP OFFER message, as normal



COMMAND SUMMARY

```
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
```

51. DYNAMIC ARP INSPECTION

WHAT IS DYNAMIC ARP INSPECTION (DAI) ?
ARP REVIEW

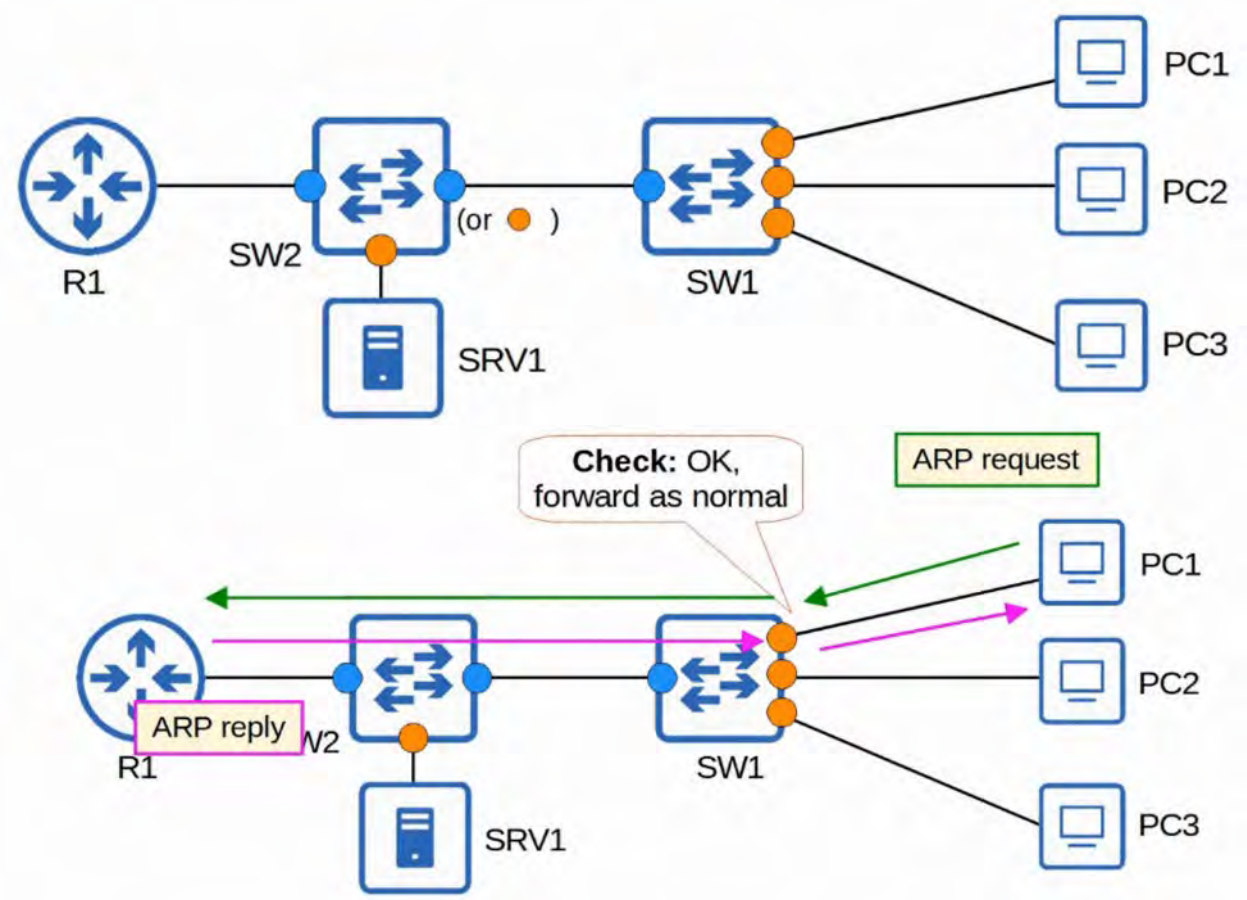
- ARP is used to learn the MAC ADDRESS of another DEVICE with a known IP ADDRESS
    - For example, a PC will use ARP to learn the MAC ADDRESS of its DEFAULT GATEWAY to communicate with external NETWORKS
- Typically, it is a TWO MESSAGE EXCHANGE : ARP REQUEST and ARP REPLY

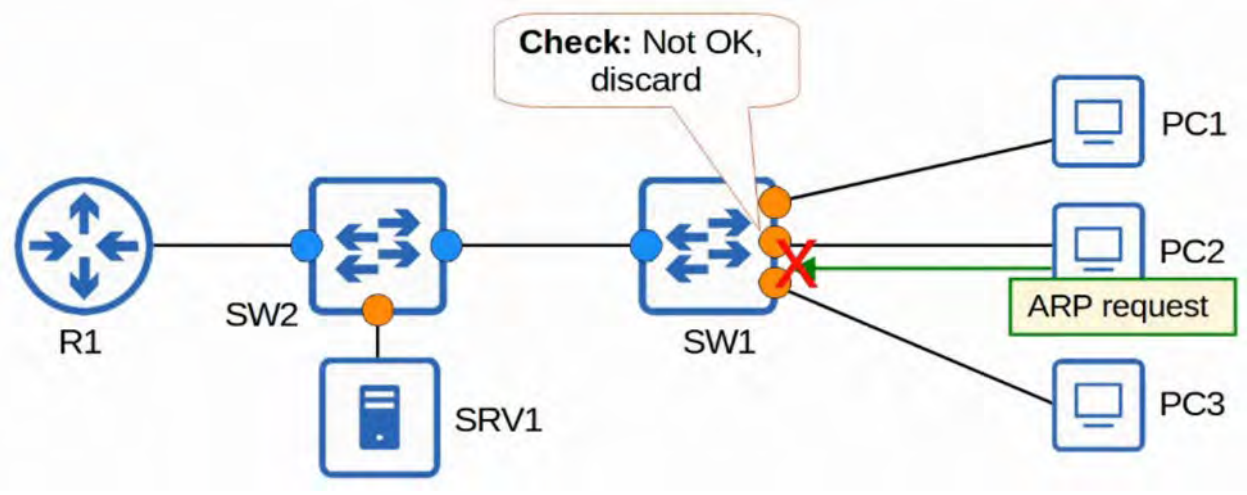
GRATUITOUS ARP

- 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

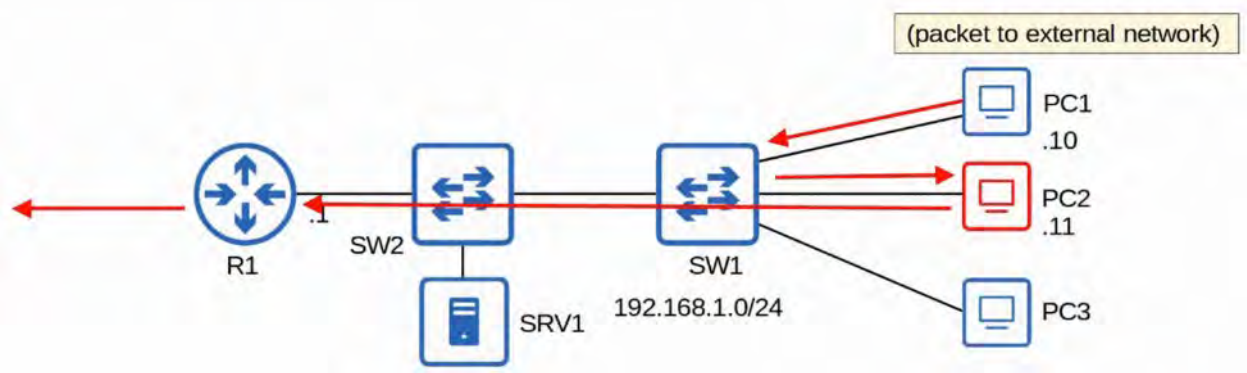
- DAI is a SECURITY FEATURE of SWITCHES that is used to filter ARP MESSAGES received on *UNTRUSTED PORTS*
- DAI only filters ARP MESSAGES. Non-ARP MESSAGES are NOT affected
- All PORTS are *UNTRUSTED*, by DEFAULT
    - Typically, all PORTS connected to other NETWORK DEVICES (SWITCHES, ROUTERS) should be configured as TRUSTED, while INTERFACES connected to END HOSTS should remain UNTRUSTED





ARP POISONING (MAN IN THE MIDDLE)

- Similar to DHCP POISONING, ARP POISONING involved an ATTACKER manipulating TARGET'S ARP TABLES so TRAFFIC is sent to the ATTACKER
- To do this, the ATTACKER can send GRATUITOUS ARP MESSAGES using another DEVICE'S IP ADDRESS
- Other DEVICES in the NETWORK will receive the GARP and update their ARP TABLES, causing them to send TRAFFIC to the ATTACKER instead of the legitimate DESTINATION



DYNAMIC ARP INSPECTION OPERATIONS

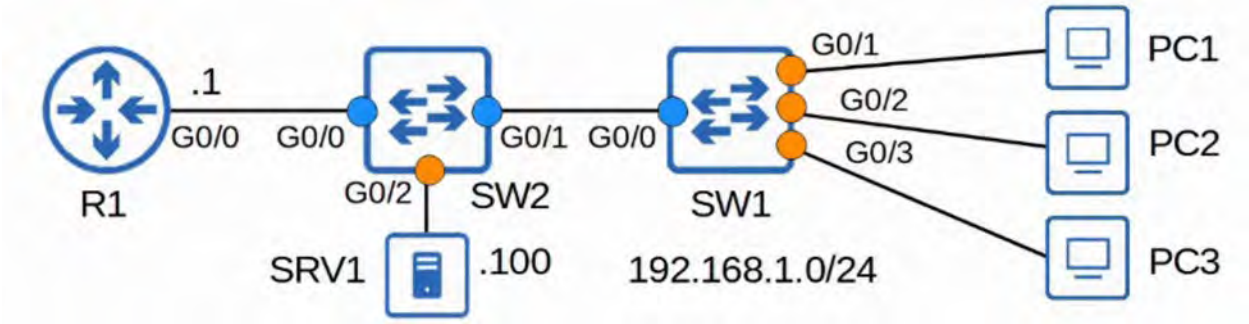
- DAI inspects the SENDER MAC and SENDER IP fields of ARP MESSAGES received on UNTRUSTED PORTS and checks that there is a matching entry in the DHCP SNOOPING BINDING TABLE
    - If there is a MATCH, the ARP MESSAGE is FORWARDED
    - If there is NO MATCH, the ARP MESSAGE is DISCARDED

```
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

DYNAMIC ARP INSPECTION CONFIGURATION




```
SW2(config)#ip arp inspection vlan 1
SW2(config)#interface range g0/0 - 1
SW2(config-if-range)#ip arp inspection trust
```

```
SW1(config)#ip arp inspection vlan 1
SW1(config)#interface g0/0
SW1(config-if)#ip arp inspection trust
```

DHCP snooping requires two commands to enable it:
ip dhcp snooping
ip dchp snooping vlan *vlan-number*
DAI only requires one:
ip arp inspection vlan *vlan-number*

Command : show ip arp inspection interfaces

```
SW1#show ip arp inspection interfaces
 Interface        Trust State       Rate (pps)     Burst Interval
 ---------------  -----------       ----------     --------------
 Gi0/0            Trusted                 None                N/A
 Gi0/1            Untrusted                 15                  1
 Gi0/2            Untrusted                 15                  1
 Gi0/3            Untrusted                 15                  1
 Gi1/0            Untrusted                 15                  1
 Gi1/1            Untrusted                 15                  1
 Gi1/2            Untrusted                 15                  1
 Gi1/3            Untrusted                 15                  1
 Gi2/0            Untrusted                 15                  1
 Gi2/1            Untrusted                 15                  1
 Gi2/2            Untrusted                 15                  1
 Gi2/3            Untrusted                 15                  1
 Gi3/0            Untrusted                 15                  1
 Gi3/1            Untrusted                 15                  1
 Gi3/2            Untrusted                 15                  1
 Gi3/3            Untrusted                 15                  1
```

DAI rate limiting is enabled on untrusted ports by default with a rate of 15 packets per second.
It is disabled on trusted ports by default.
*DHCP snooping rate limiting is disabled on all interfaces by default.

DHCP snooping rate limiting is configured like this:
x packets per second.

The DAI **burst interval** allows you to configure rate limiting like this:
x packets per y seconds

DAI RATE LIMITING

```
SW1(config)#interface range g0/1 - 2
SW1(config-if-range)#ip arp inspection limit rate 25 burst interval 2
SW1(config-if-range)#interface range g0/3
SW1(config-if)#ip arp inspection limit rate 10
SW1(config-if)#do show ip arp inspection interfaces

 Interface        Trust State       Rate (pps)     Burst Interval
 ---------------  -----------       ----------     --------------
 Gi0/0            Trusted                 None                N/A
 Gi0/1            Untrusted                 25                  2
 Gi0/2            Untrusted                 25                  2
 Gi0/3            Untrusted                 10                  1
![output omitted]

SW1(config)#errdisable recovery cause arp-inspection
SW1(config)#do show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
arp-inspection               Enabled
![outputed omitted]
```

The burst interval is optional. If you don't specify it, the default is 1 second.

If ARP messages are received faster than the specified rate, the interface will be err-disabled.
It can be re-enabled in two ways:
1: **shutdown** and **no shutdown**
2: **errdisable recovery cause arp-inspection**

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)

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

```
> Frame 224: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
v Ethernet II, Src: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00), Dst: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  > Destination: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  > Source: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
    Sender IP address: 192.168.1.1
    Target MAC address: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    Target IP address: 192.168.1.10
```

These checks are done in addition to the standard DAI check (sender MAC/IP).
If configured, an ARP message must pass **all** of the checks to be considered valid.

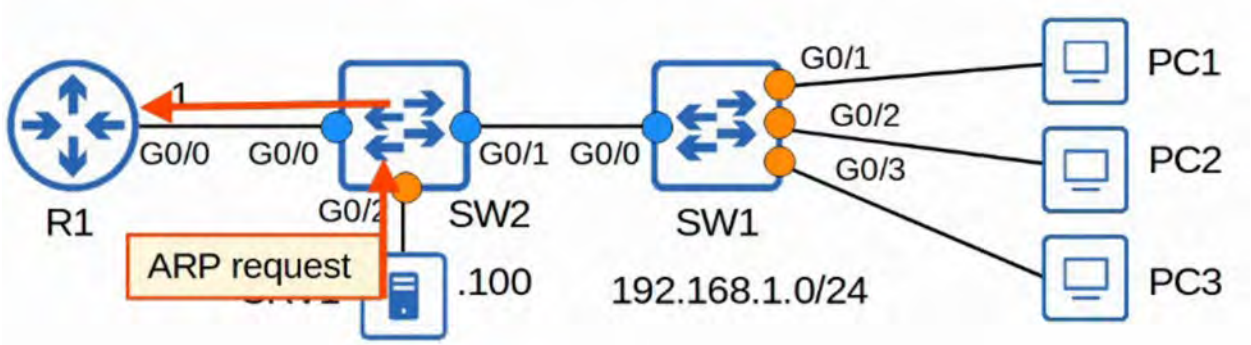
```
SW1(config)#ip arp inspection validate dst-mac
SW1(config)#ip arp inspection validate ip
SW1(config)#ip arp inspection validate src-mac

SW1(config)#do show running-config | include validate
ip arp inspection validate src-mac

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

SW1(config)#do show running-config | include validate
ip arp inspection validate src-mac dst-mac ip
```

You must enter all of the validation checks you want in a single command.
*You can specify one, two, or all three.
*The order isn't significant.

ARP ACLs (Beyond Scope of CCNA)
CREATE AN ARP ACL FOR SRV1

```
SW2#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
0C:29:2F:18:79:00   192.168.1.12     79226       dhcp-snooping   1     GigabitEthernet0/1
0C:29:2F:90:91:00   192.168.1.10     79188       dhcp-snooping   1     GigabitEthernet0/1
0C:29:2F:67:E9:00   192.168.1.11     79210       dhcp-snooping   1     GigabitEthernet0/1
Total number of bindings: 3

!SRV1 has a static IP address of 192.168.1.100, so it does not have an entry in SW2's DHCP
!snooping binding table.

*Jun 19 05:56:15.538: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Gi0/2, vlan 1.
([0c29.2f1e.7700/192.168.1.100/0000.0000.0000/192.168.1.1/05:56:14 UTC Sat Jun 19 2021])

SW2(config)#arp access-list ARP-ACL-1
SW2(config-arp-nacl)#permit ip host 192.168.1.100 mac host 0c29.2f1e.7700

SW2(config)#ip arp inspection filter ARP-ACL-1 vlan 1
```

AFTER APPLYING IT TO SWITCH 2, SRV1 is able to send ARP REQUEST to R1



Command: show ip arp inspection
Shows a summary of the DAI configuration and statistics

```
SW2#show ip arp inspection

Source Mac Validation      : Enabled
Destination Mac Validation : Enabled
IP Address Validation      : Enabled

 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
   1      Enabled          Active      ARP-ACL-1          No

 Vlan     ACL Logging      DHCP Logging      Probe Logging
 ----     -----------      ------------      -------------
   1      Deny             Deny              Off

 Vlan      Forwarded        Dropped     DHCP Drops      ACL Drops
 ----      ---------        -------     ----------      ---------
   1            56              4              4              0

 Vlan   DHCP Permits    ACL Permits   Probe Permits   Source MAC Failures
 ----   ------------    -----------   -------------   -------------------
   1             0              1               0                     0

 Vlan   Dest MAC Failures   IP Validation Failures   Invalid Protocol Data
 ----   -----------------   ----------------------   ---------------------

 Vlan   Dest MAC Failures   IP Validation Failures   Invalid Protocol Data
 ----   -----------------   ----------------------   ---------------------
   1                    0                        0                      0
```

- If **static ACL** is set to **yes**, the implicit deny at the end of the ARP ACL will take effect.
- This will cause all ARP messages not permitted by the ARP ACL to be denied.
- In effect, this means that only the ARP ACL will be checked, the DHCP snooping table will not be checked.

COMMAND REVIEW

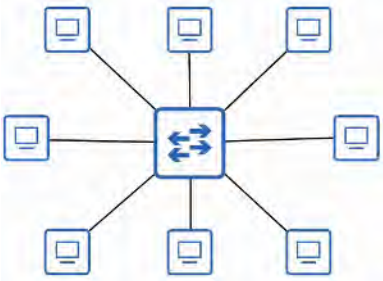
```
SW1(config)# ip arp inspection vlan vlan-number
SW1(config)# errdisable recovery cause arp-inspection
SW1(config)# ip arp inspection validate (src-mac | dst-mac | ip)
SW1(config-if)# ip arp inspection trust
SW1(config-if)# ip arp inspection limit rate packets [burst interval seconds]

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
```

52. LAN ARCHITECTURES

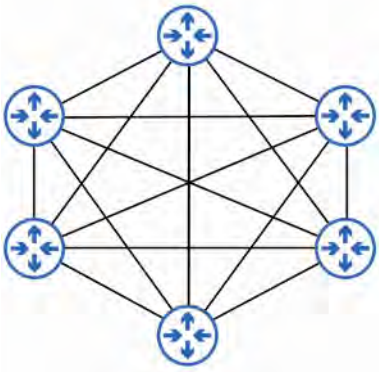
- You have studied various NETWORK technologies: ROUTING, SWITCHING, STP, ETHERCHANNEL, OSPF, FHRPs, SWITCH SECURITY FEATURES, etc.
    - Now, let's look at some BASIC NETWORK DESIGN / ARCHITECTURE
- There are standard "BEST PRACTICES" for NETWORK DESIGN
    - However there are a few UNIVERSAL "CORRECT ANSWERS"
    - The answer to MOST general questions about NETWORK DESIGN is "IT DEPENDS"
- In the early stages of your NETWORKING career, you probably won't be asked to DESIGN NETWORKS yourself
- However, to understand the NETWORKS you will be CONFIGURING and TROUBLESHOOTING, it's important to know some BASICS of NETWORK DESIGN

COMMON TERMINOLOGIES

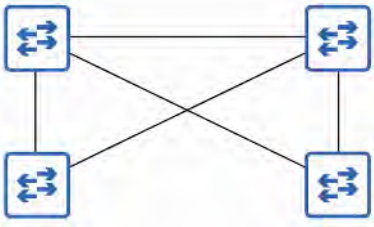
- STAR
    - When several DEVICES all connect to ONE CENTRAL DEVICE, we can draw them in a "STAR" shape like below, so this is often called a "STAR TOPOLOGY"



- FULL MESH
    - When each DEVICE is connected to each OTHER DEVICE



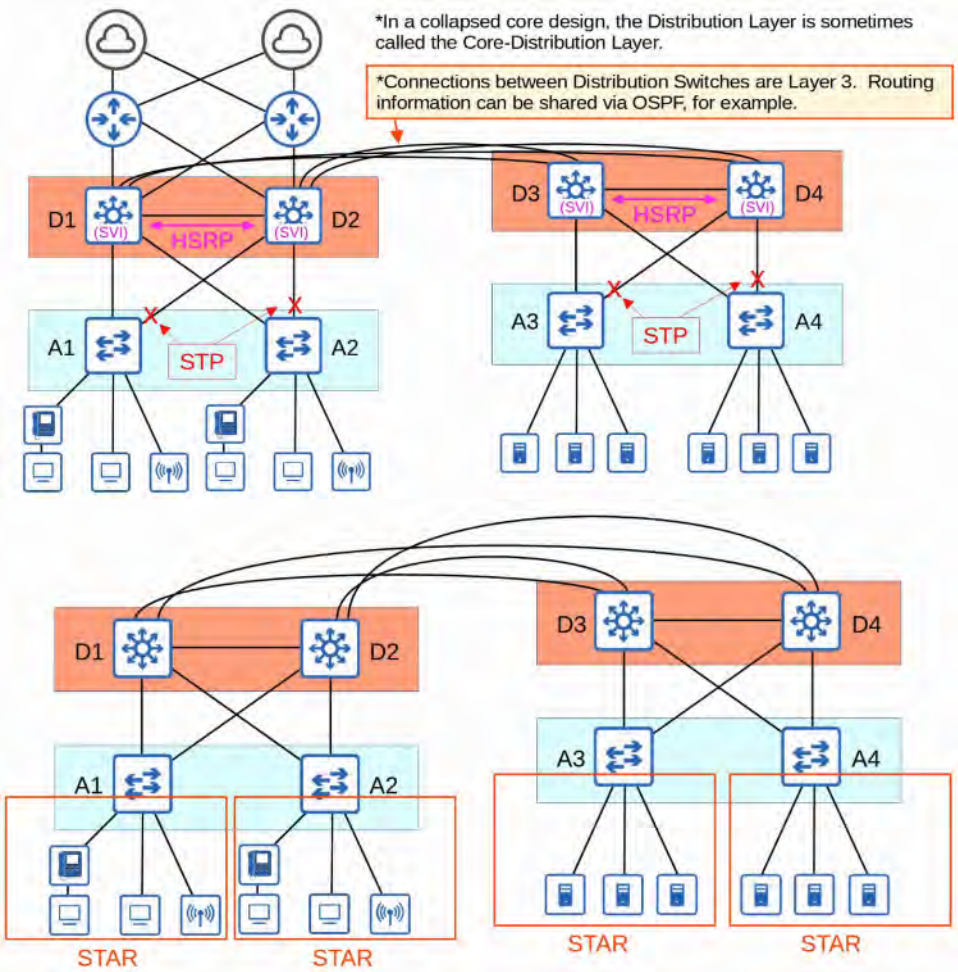
- PARTIAL MESH
    - When SOME DEVICES are connected to each other but not ALL

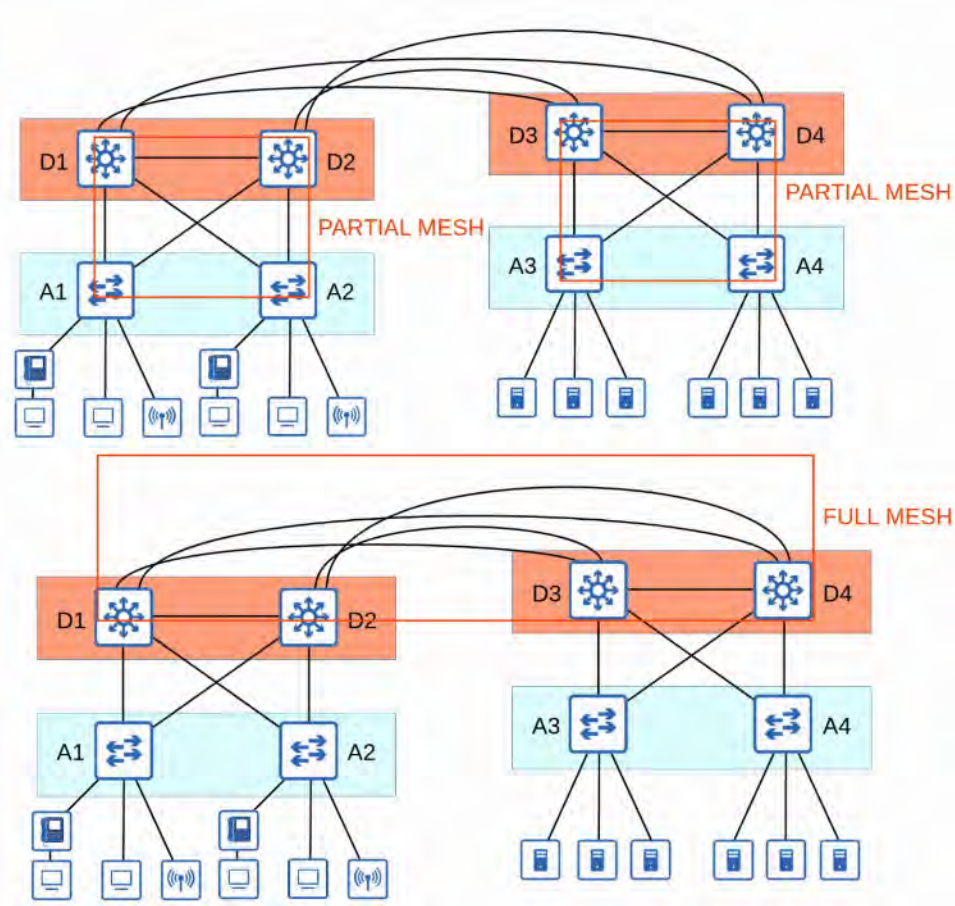


2-TIER AND 3-TIER LAN ARCHITECTURE

- The TWO-TIER LAN DESIGN consists of TWO Hierarchical Layers:
    - ACCESS LAYER

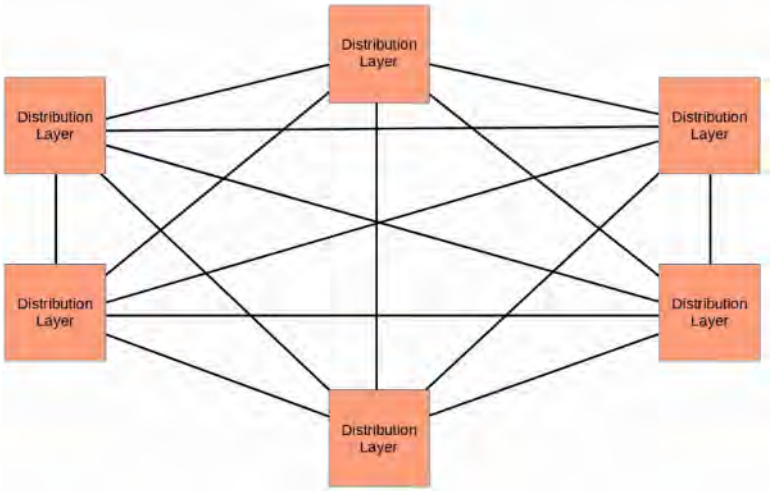
    - DISTRIBUTION LAYER
- Also called a “COLLAPSED CORE” DESIGN because it omits a layer that is found in the THREE TIER DESIGN : THE CORE LAYER
- ACCESS LAYER
    - The LAYER that END HOSTS connect to (PCs, Printers, Cameras, etc)
    - Typically, ACCESS LAYER SWITCHES have lots of PORTS for END HOSTS to connect to
    - QoS MARKING is typically done here
    - Security Services like PORT SECURITY, DAI, etc are typically performed here
    - SWITCHPORTS might be PoE-Enabled for Wireless APs, IP Phones, etc.
- DISTRIBUTION LAYER
    - Aggregates connections from the ACCESS LAYER SWITCHES
    - Typically is the border between LAYER 2 and LAYER 3
    - Connects to services such as Internet, WAN, etc
    - Sometimes called AGGREGATION LAYER





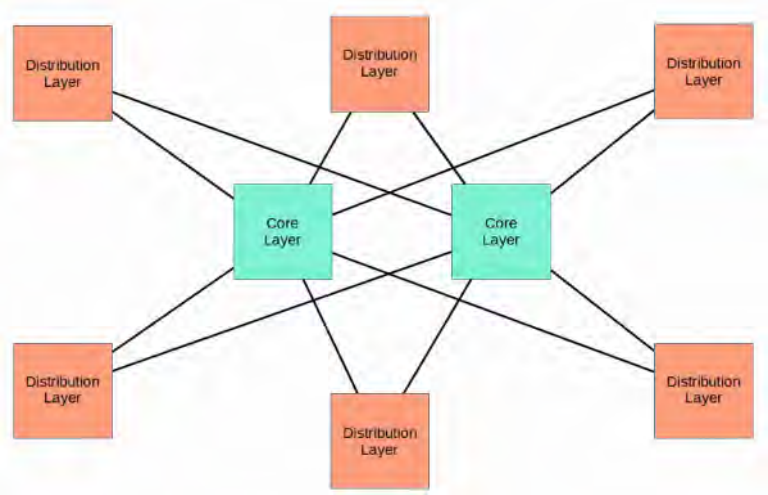
THREE-TIER CAMPUS LAN DESIGN

- In large NETWORKS with many DISTRIBUTION LAYER SWITCHES (for example in separate buildings), the number of connections required between DISTRIBUTION LAYER SWITCHES grows rapidly

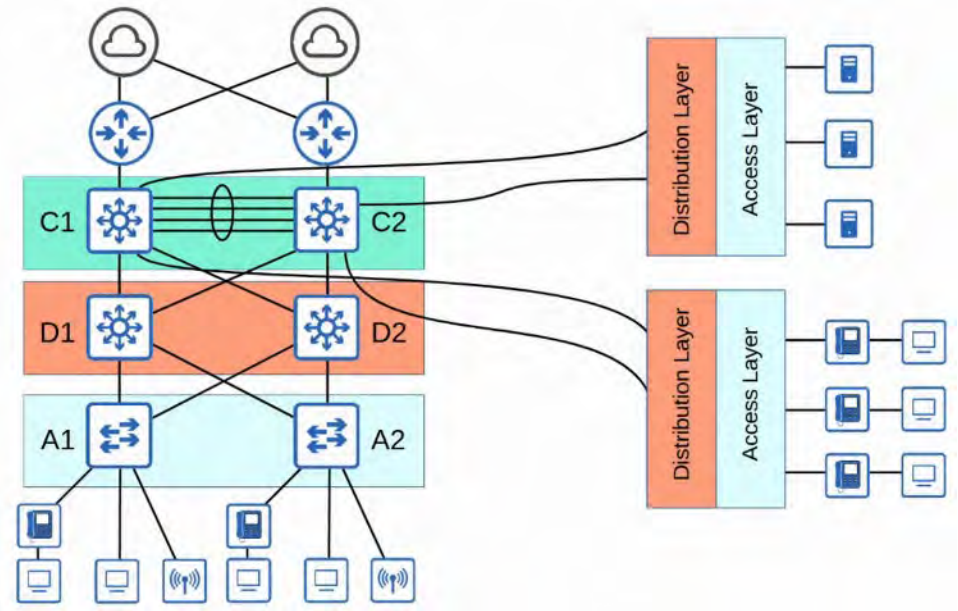


- To help SCALE large LAN NETWORKS, you can add a CORE LAYER.

** Cisco recommends adding a CORE LAYER if there are more than THREE DISTRIBUTION LAYERS in a single location

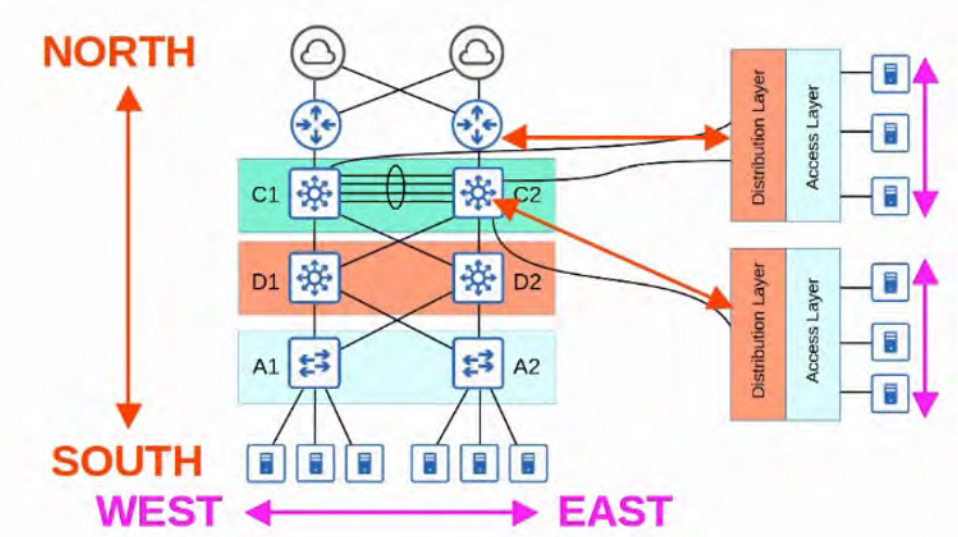


- The THREE-TIER LAN DESIGN consists of THREE HIERARCHICAL LAYERS:
  - ACCESS LAYER
  - DISTRIBUTION LAYER
  - CORE LAYER
- CORE LAYER:
  - Connects DISTRIBUTION LAYERS together in large LAN NETWORKS
  - The focus is SPEED (”FAST TRANSPORT”)
  - CPU-INTENSIVE OPERATIONS, such as SECURITY, QoS Markings / Classification, etc. should be avoided at this LAYER
  - Connections are all LAYER 3. NO SPANNING-TREE!
  - Should maintain connectivity throughout the LAN even if DEVICES FAIL



SPINE-LEAF ARCHITECTURE (DATA CENTER)

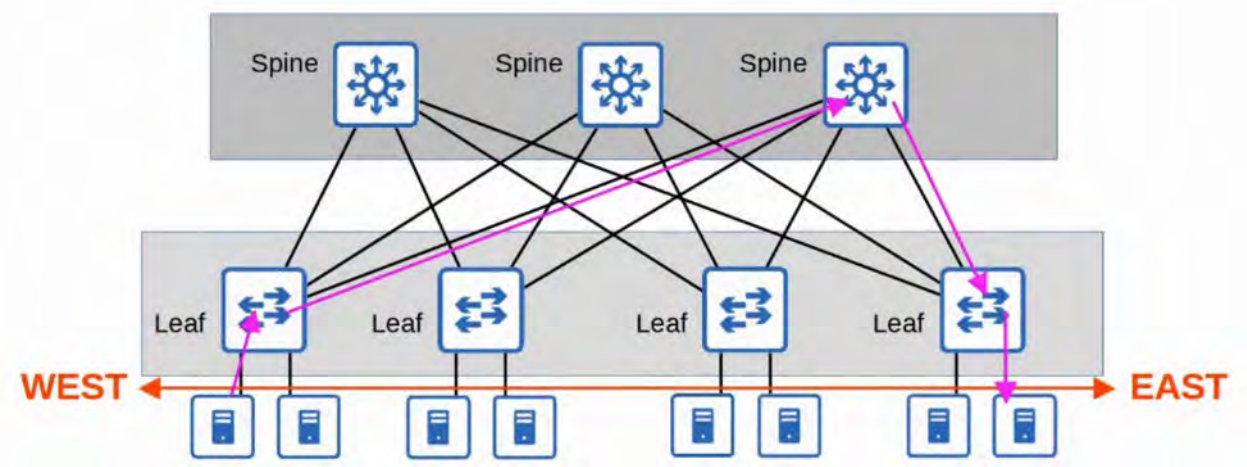
- CISCO ACI ARCHITECTURE (Application Centric Infrastructure) uses this architecture
- DATA CENTERS are dedicated spaces / buildings used to STORE COMPUTER SYSTEMS such as SERVERS and NETWORK DEVICES
- Traditional DATA CENTER designs used a THREE-TIER ARCHITECTURE (ACCESS-DISTRIBUTION-CORE) like we just covered
- This worked well when most TRAFFIC in the DATA CENTER was NORTH-SOUTH



- With the precedence of VIRTUAL SERVERS, applications are often deployed in a DISTRIBUTED manner (across multiple physical SERVERS) which increases the amount of EAST-WEST TRAFFIC in the DATA CENTER
- The traditional THREE-TIER ARCHITECTURE led to bottlenecks in the BANDWIDTH as well as VARIABILITY in the SERVER-TO-SERVER latency depending on the PATH the TRAFFIC takes
- To SOLVE this, SPINE-LEAF ARCHITECTURE (also called CLOS ARCHITECTURE) has become prominent in DATA CENTERS

RULES FOR SPINE-LEAF ARCHITECTURE

- Every LEAF SWITCH is connected to every SPINE SWITCH
- Every SPINE SWITCH is connected to every LEAF SWITCH
- LEAF SWITCHES do NOT connect to other LEAF SWITCHES
- SPINE SWITCHES do NOT connect to other SPINE SWITCHES
- END HOSTS (Servers, etc) ONLY connect to LEAF SWITCHES

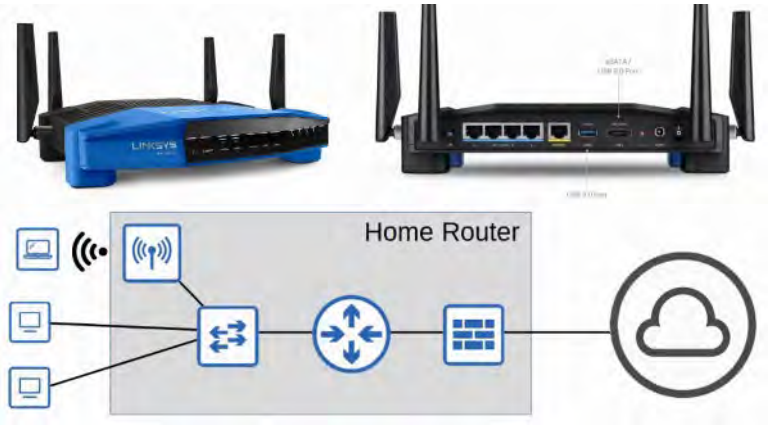


- The PATH taken by TRAFFIC is randomly chosen to balance the TRAFFIC LOAD among the SPINE SWITCHES
- Each SERVER is separated by the same number of “HOPS” (except those connected to the same LEAF) providing CONSISTENT LATENCY for EAST-WEST TRAFFIC

SOHO (SMALL OFFICE / HOME OFFICE)

- SMALL OFFICE / HOME OFFICE (SOHO) refers to the office of a small company, or a small home office with few DEVICES

    - Doesn’t have to be an actual home “office”; if your home has a NETWORK connected to the INTERNET it is considered a SOHO NETWORK
- SOHO NETWORKS don’t have complex needs, so all NETWORKING functions are typically provided by a SINGLE DEVICE, often called a “HOME ROUTER” or “WIRELESS ROUTER”
- The one DEVICE can serve as a:
    - ROUTER
    - SWITCH
    - FIREWALL
    - WIRELESS ACCESS POINT
    - MODEM



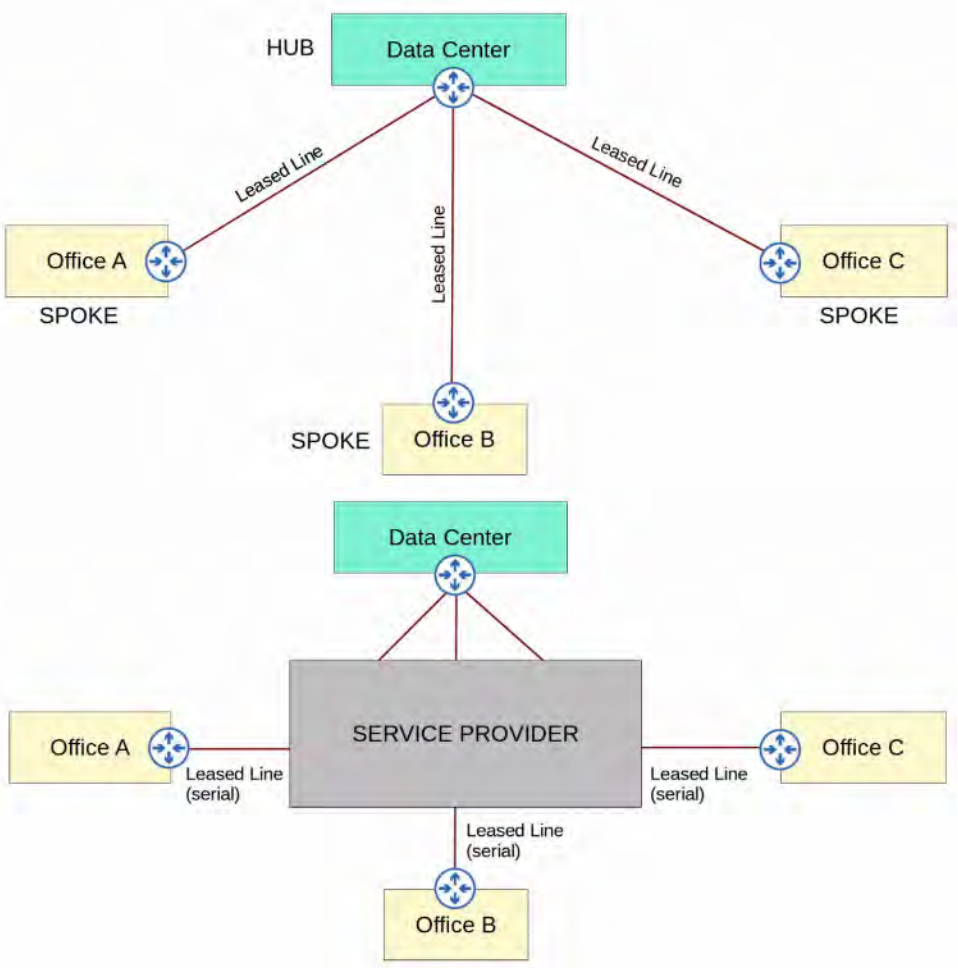
53. WAN ARCHITECTURES

INTRODUCTION TO WANS

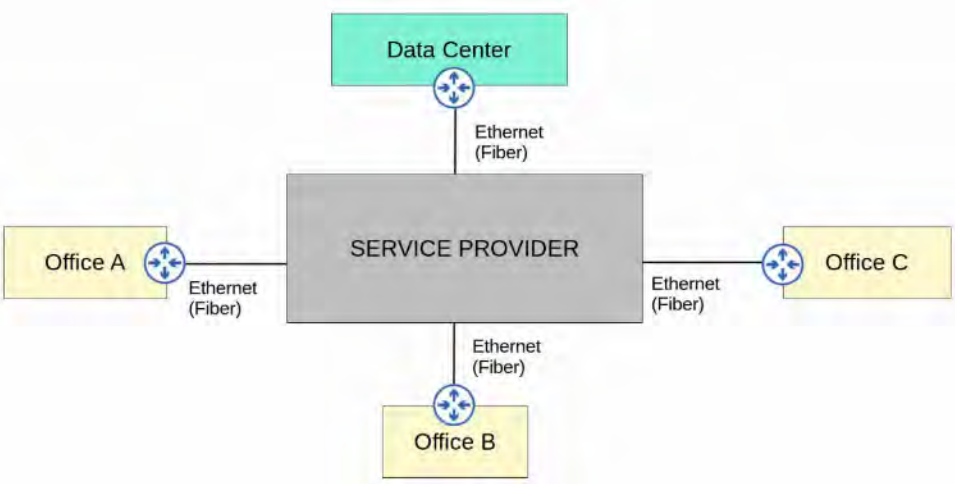
- WAN stands for WIDE AREA NETWORK
- A WAN is a NETWORK that extends over a large geographic area
- WANs are used to connect geographically separate LANs
- Although the Internet can be considered a WAN, the term “WAN” is typically used to refer to an enterprise’s private connections that connect their offices, data centers, and other sites together
- Over public/shared networks like the Internet, VPNs (Virtual Private Networks) can be used to create private WAN connections
- There have been many different WAN technologies over the years. Depending on the location, some will be available and some will not be
- Technologies which are considered “legacy” (old) in one country, might still be used in other countries

---

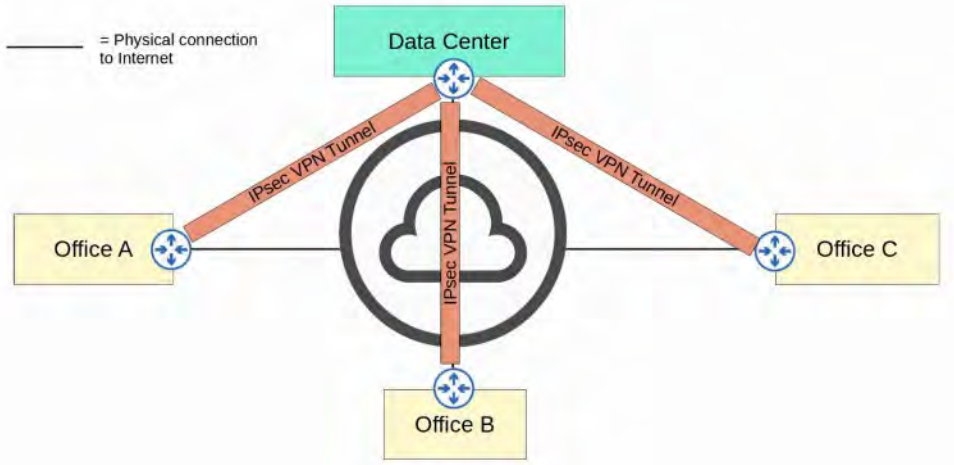
WAN OVER DEDICATED CONNECTION (LEASED LINE)
HUB-and-SPOKE topology



WAN CONNECTION VIA ETHERNET (FIBER)



WAN OVER SHARED INFRASTRUCTURE (INTERNET VPN)



LEASED LINES

- A LEASED LINE is a dedicated physical link, typically connecting two sites
- LEASED LINES use serial connections (PPP or HDLC encapsulation)
- There are various standards that provide different speeds and different standards are available in different countries.
- Due to the HIGHER cost, HIGHER installation lead time, and SLOWER speeds of LEASED LINES, Ethernet WAN technologies are becoming MORE popular

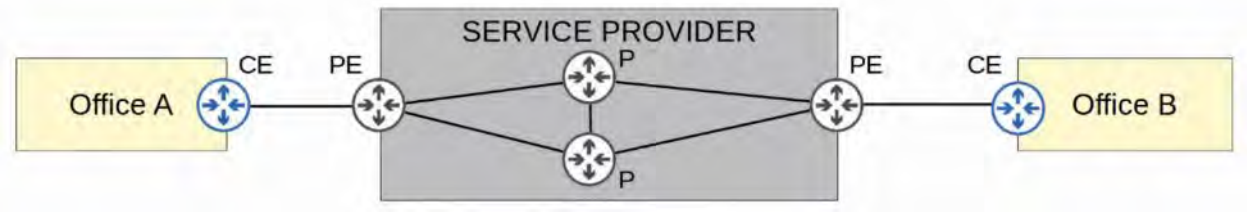
System	North American	Japanese	European (CEPT)
Level zero (channel data rate)	64 kbit/s (DS0)	64 kbit/s	64 kbit/s
First level	1.544 Mbit/s (DS1) (24 user channels) (T1)	1.544 Mbit/s (24 user channels)	2.048 Mbit/s (32 user channels) (E1)
(Intermediate level, T-carrier hierarchy only)	3.152 Mbit/s (DS1C) (48 Ch.)	–	–
Second level	6.312 Mbit/s (DS2) (96 Ch.) (T2)	6.312 Mbit/s (96 Ch.), or 7.786 Mbit/s (120 Ch.)	8.448 Mbit/s (128 Ch.) (E2)
Third level	44.736 Mbit/s (DS3) (672 Ch.) (T3)	32.064 Mbit/s (480 Ch.)	34.368 Mbit/s (512 Ch.) (E3)
Fourth level	274.176 Mbit/s (DS4) (4032 Ch.)	97.728 Mbit/s (1440 Ch.)	139.264 Mbit/s (2048 Ch.) (E4)
Fifth level	400.352 Mbit/s (DS5) (5760 Ch.)	565.148 Mbit/s (8192 Ch.)	565.148 Mbit/s (8192 Ch.) (E5)

Wikipedia: 'Comparison of T-carrier and E-carrier systems'

MPLS VPNs

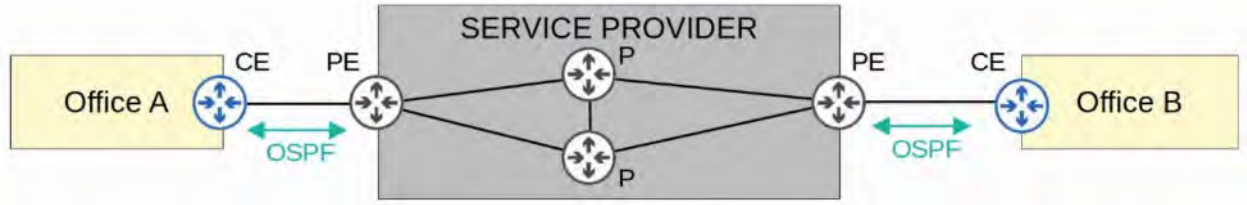
- MPLS stands for "Multi Protocol Label Switching"
- Similar to the Internet, service providers' MPLS NETWORKS are shared infrastructure because many customer enterprises connect to and share the same infrastructure to make WAN connections

- However, the “label switching” in the name of MPLS allows VPNs to be created over the MPLS infrastructure through the use of LABELS
- IMPORTANT terms:
    - CE ROUTER = Customer Edge ROUTER
    - PE ROUTER = Provider Edge ROUTER
    - P ROUTER = Provider Core ROUTER



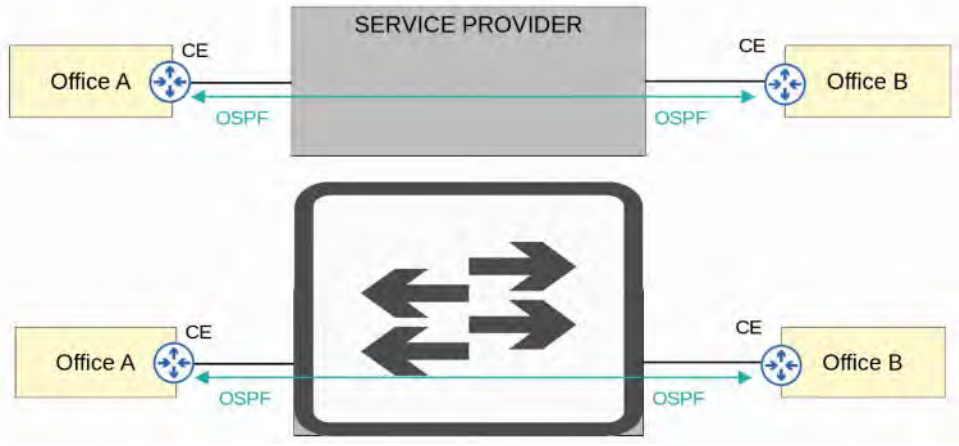
- When the PE ROUTERS receive FRAMES from the CE ROUTERS, they add a LABEL to the FRAME
- These LABELS are used to make forwarding decisions within the SERVICE PROVIDER NETWORK - NOT the DESTINATION IP
- The CE ROUTERS do NOT USE MPLS, it is only used by the PE/P ROUTERS
- When using a LAYER 3 MPLS VPN, the CE and PE ROUTERS peer using OSPF, for example, to share ROUTING information

EXAMPLE:
OFFICE A’s CE will peer with one PE
OFFICE B’s CE will peer with the other PE
OFFICE A’s CE will learn about OFFICE B’s ROUTES via this OSPF peering
OFFICE B’s CE will learn about OFFICE A’s ROUTES as well

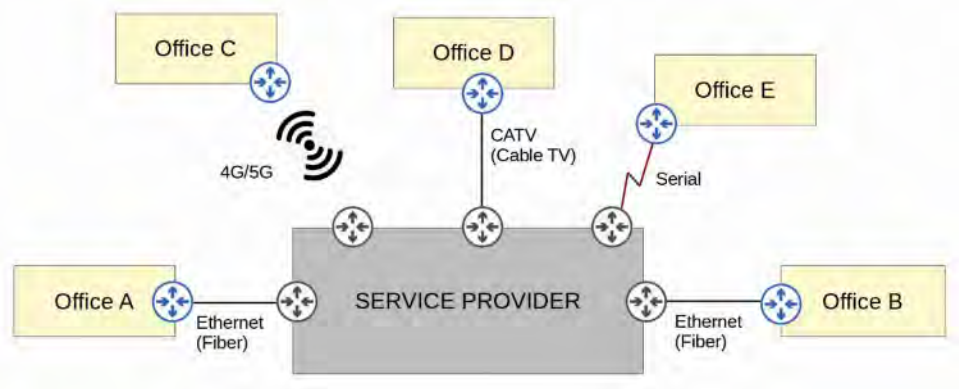


- When using a LAYER 2 MPLS VPN, the CE and PE ROUTERS do NOT form PEERINGS
- The SERVICE PROVIDER NETWORK is entirely *transparent* to the CE ROUTERS
- In effect, it is like the TWO CE ROUTERS are directly connected.
    - Their WAN INTERFACES will be in the SAME SUBNET
- If a ROUTING protocol is used, the TWO CE ROUTERS will peer directly with each other

CE ROUTERS connected via LAYER 2 MPLS VPN



- Many different technologies can be used to connect to a SERVICE PROVIDER's MPLS NETWORK for WAN Service

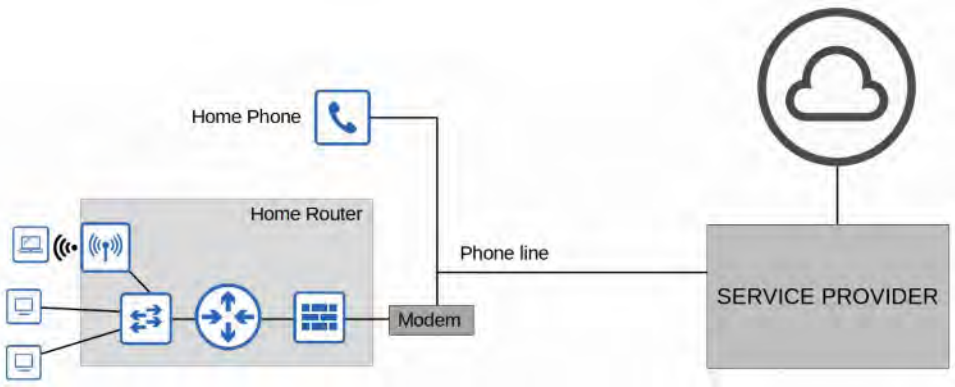


INTERNET CONNECTIVITY

- There are countless ways for an enterprise to connect to the INTERNET
- For example, PRIVATE WAN technologies such as LEASED LINES and MPLS VPNs can be used to connect to a SERVICE PROVIDER's INTERNET infrastructure
- In addition, technologies such as CATV and DSL commonly used by consumers (Home Internet Access) can also be used by an enterprise
- These days for both enterprise and consumer INTERNET access, FIBER OPTIC ETHERNET connections are growing in popularity due to high speeds they provide over long distances
- Let's briefly look at TWO INTERNET access technologies mentioned above:
    - CABLE (CATV)
    - DSL

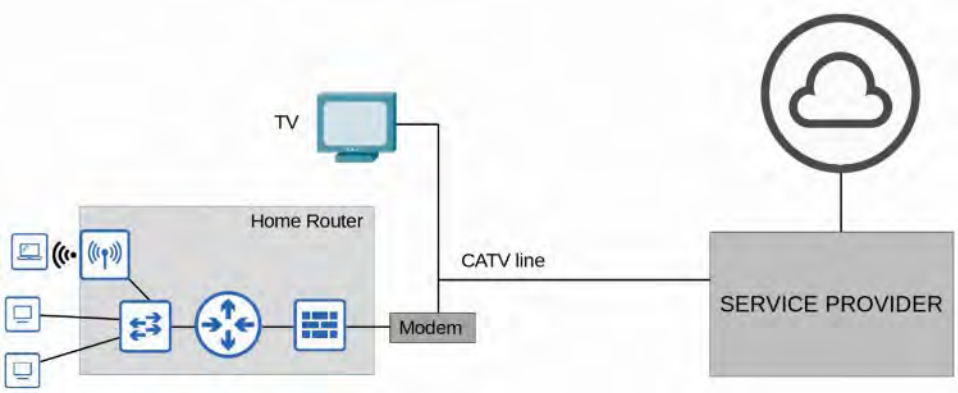
DIGITAL SUBSCRIBER LINE (DSL)

- DSL provides INTERNET connectivity to customers over phone lines and can share the same phone line that is already installed in most homes
- A DSL MODEM (Modulator / Demodulator) is required to convert DATA into a format suitable to be sent over the phone lines
    - The MODEM might be a separate DEVICE or it might be incorporated in to a "HOME ROUTER"

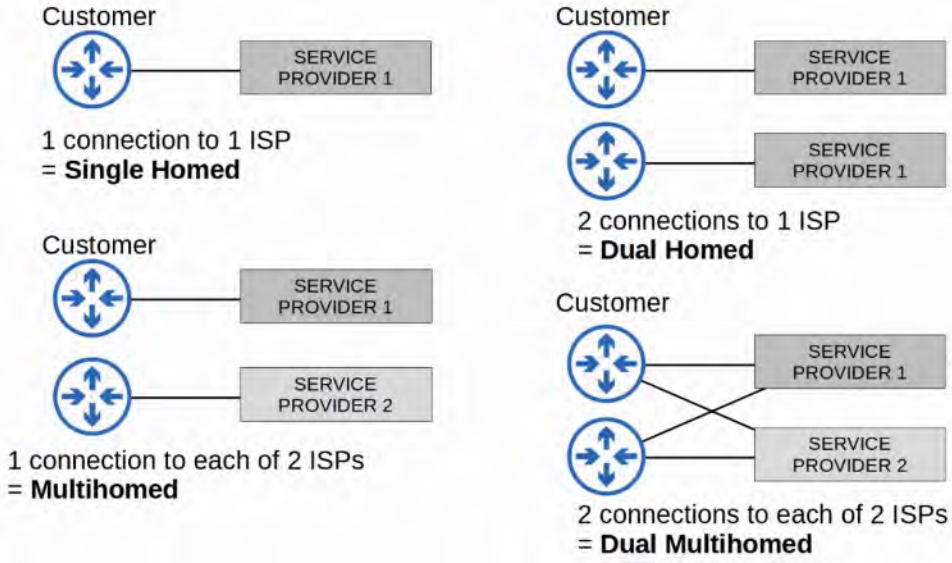


CABLE INTERNET

- CABLE INTERNET provides INTERNET ACCESS via the same CATV (Cable Television) lines used for TV service
- Like DLS, a CABLE MODEM is required to convert DATA into a format suitable to be sent over the CATV CABLES.
    - Like a DSL MODEM, this can be a separate device or built into the HOME ROUTER



REDUNDANT INTERNET CONNECTIONS

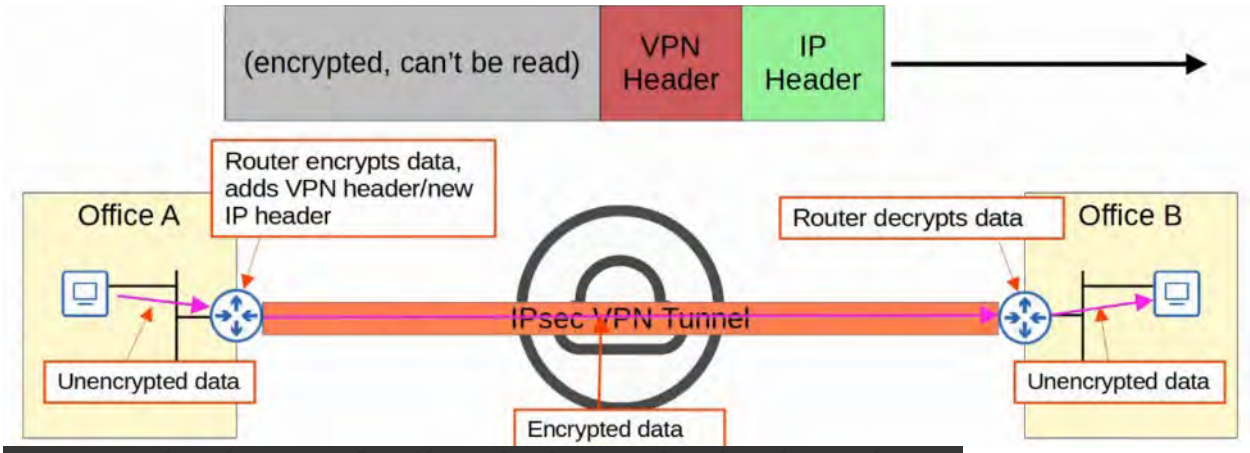


INTERNET VPNs

- PRIVATE WAN SERVICES such as LEASED LINES and MPLS provide security because each customer's TRAFFIC is separated by using dedicated physical connections (LEASED LINE) or by MPLS TAGS
- When using the INTERNET as a WAN to connect SITES together, there is no built-in security by DEFAULT
- To provide secure communications over the Internet, VPNs (Virtual Private Networks) are used
- We will cover two kinds of Internet VPNs:
    - SITE-TO-SITE VPNS using IPSec
    - REMOTE-ACCESS VPNs using TLS

SITE-TO-SITE VPNs (IPSec)

- A "SITE-TO-SITE" VPN is a VPN between two DEVICES and is used to connect TWO SITES together over the INTERNET
- A VPN "TUNNEL" is created between the TWO DEVICES by ENCAPSULATING the original IP PACKET with a VPN HEADER and a new IP HEADER
    - When using IPSec, the original PACKET is encrypted before its ENCAPSULATED with the new HEADER



Internet Protocol Security (IPSec) configured in tunnel mode encrypts the entire packet. IPSec is a suite of protocols that can be used to encrypt Generic Routing Encapsulation (GRE) tunnel traffic, such as over a virtual private network (VPN). IPSec supports two modes: transport and tunnel. In tunnel mode, IPSec encrypts the entire Internet Protocol (IP) packet, including the header.

IPSec configured in tunnel mode does require an additional header. This is because the entire packet is encrypted. Layer 3 forwarding devices are not capable of decrypting the encrypted packet. Therefore, a new unencrypted IP header encapsulates the encrypted packet for routing.

IPSec configured in transport mode, not tunnel mode, does not encrypt the IP header. In transport mode, only the IP packet's payload is encrypted by IPSec, which means that the IP packet's header remains intact.

IPSec tunnel mode is not required for Network Address Translation (NAT) traversal. However, IPSec tunnel mode is more compatible with NAT than IPSec transport mode. IPSec in tunnel mode encrypts the entire packet, including both the header and payload. The encrypted packet is then encapsulated in a new IP packet that includes a new IP header, which can be used by NAT. Transport mode, on the other hand, creates complications for NAT if Authentication Header (AH) is used. It is possible to use IPSec in transport mode when NAT is deployed. However, it requires the use of a NAT Traversal (NAT-T) solution as described in the Internet Engineering Task Force (IETF) Request for Comments (RFC) 3947.

PROCESS SUMMARY:

1. The SENDING DEVICE combines the original PACKET and SESSION KEY (ENCRYPTION KEY) and runs them through an ENCRYPTION FORMULA
2. The SENDING DEVICE encapsulates the ENCRYPTED PACKET with a VPN HEADER and a new IP HEADER
3. The SENDING DEVICE sends the NEW PACKET to the DEVICE on the other side of the TUNNEL
4. The RECEIVING DEVICE decrypts the DATA to get the original PACKET and then forwards the original PACKET to it's DESTINATION

- In a "SITE-TO-SITE" VPN, a TUNNEL is formed only between TWO TUNNEL ENDPOINTS (for example, the TWO ROUTERS connected to the INTERNET)
- All OTHER DEVICES in each site DO NOT need to create a VPN for themselves. They can send unencrypted DATA to their site's ROUTER, which will ENCRYPT it and FORWARD it in the TUNNEL as described above.

LIMITATIONS OF STANDARD IPSec

1. IPSec doesn't support BROADCAST or MULTICAST TRAFFIC, only UNICAST.

- This means that ROUTING PROTOCOLS such as OSPF cannot be used over the TUNNELS because they rely on MULTICAST TRAFFIC

    - This can be SOLVED with “GRE over IPSec”
2. Configuring a full mesh of TUNNELS between many sites is a labor-intensive task

Let’s look at each of the above SOLUTIONS

GRE over IPSec

- GRE (GENERIC ROUTING ENCAPSULATION) creates TUNNELS like IPSec, however it does not ENCRYPT the original PACKET, so it is NOT SECURE
- However, it has the advantage of being able to encapsulate a WIDE variety of a LAYER 3 PROTOCOLS as well as BROADCAST and MULTICAST messages
- To get the FLEXIBILITY of GRE with the SECURITY of IPSec, “GRE over IPSec” can be used
- The original PACKET will be ENCAPSULATED by a GRE HEADER and a new IP HEADER, and then the GRE PACKET will be ENCRYPTED and ENCAPSULATED within an IPSec VPN HEADER and a NEW IP HEADER

After a sending host on a site-to-site virtual private network (VPN) that is constructed by using Generic Routing Encapsulation (GRE) with Internet Protocol Security (IPSec) for transport adds a VPN header and an Internet Protocol (IP) header to the packet, the sending host sends the packet to the destination. The addition of the VPN header and IP header to the packet is a process known as encapsulation.

A site-to-site VPN uses IPSec to transport information across a tunnel that is established between two hosts. A typical site-to-site VPN uses GRE with confidentiality, integrity, and antireplay protection provided by IPSec. There are four steps in the site-to-site VPN IPSec encryption process. By contrast, a remote access VPN uses client software to encrypt traffic between a remote user and internal company resources.

First, the sending device combines a session key, which is also known as an encryption key or a shared key, with the data that is to be transported over the tunnel. It then uses the session key to encrypt both the data and the key.

Second, the sending device encapsulates the encrypted data and session key into a packet with a VPN header and a new IP header. These headers contain the source and destination information that is used to transport the encrypted data and session key over the tunnel.

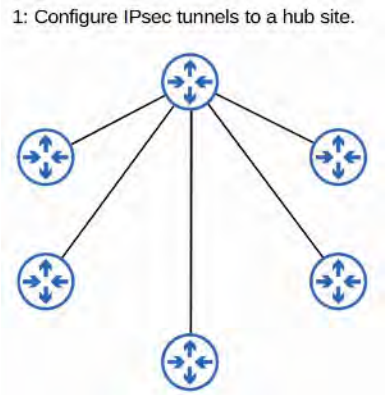
Third, the sending device sends the completed packet to the destination device at the other end of the tunnel, or site-to-site VPN.

Fourth and finally, the destination device, or receiving device, uses the same session key that the sending device used for encryption to decrypt the encrypted packet and session key.

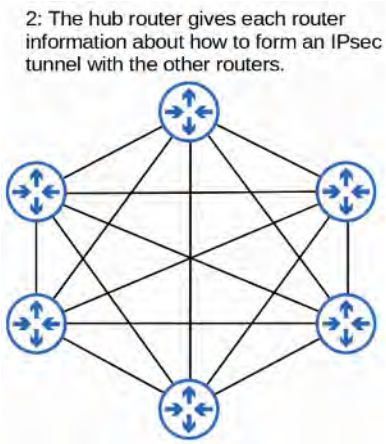
DMVPN

- DMVPN (Dynamic Multipoint VPN) is a Cisco-Developed solution that allows ROUTERS to dynamically create a FULL MESH of IPSec TUNNELS without having to manually configure every SINGLE TUNNEL

1. CONFIGURE IPSec TUNNELS to a HUB SITE



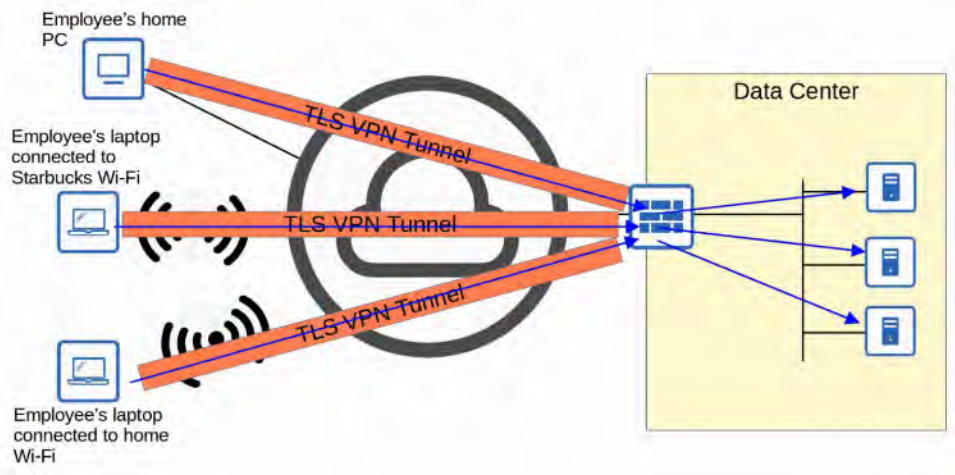
2. The HUB ROUTER gives each ROUTER information about HOW to form an IPSec TUNNEL with the OTHER ROUTERS



DMVPN provides the configuration simplicity of HUB-AND-SPOKE (each SPOKE ROUTER only needs one TUNNEL configured) and the EFFICIENCY of DIRECT SPOKE-TO-SPOKE communication (SPOKE ROUTERS can communicate directly without TRAFFIC passing through the HUB)

REMOTE-ACCESS VPNs

- Whereas SITE-TO-SITE VPNs are used to make a POINT-TO-POINT connection between TWO SITES over the INTERNET, REMOTE-ACCESS VPNs are used to allow END DEVICES (PCs, Mobile Phone) to ACCESS the company's internal resources securely over the INTERNET
- REMOTE-ACCESS VPNs typically use TLS (TRANSPORT LAYER SECURITY)
    - TLS is also what provides security for HTTPS (HTTP SECURE)
    - TLS was formerly known as SSL (Secure Socket Layer) and developed by Netscape, but it was renamed to TLS when it was standardized by the IETF
- VPN client software (for example Cisco AnyConnect) is installed on END DEVICES (for example company-provided laptops that employees use to work from home)
- These END DEVICES then form SECURE TUNNELS to one of the company's ROUTERS / FIREWALLS acting as a TLS SERVER
- This allows the END USERS to securely access RESOURCES on the company's INTERNAL NETWORK without being directly connected to the company NETWORK



SITE-TO-SITE versus REMOTE-ACCESS VPN

- SITE-TO-SITE VPNs typically use IPSec
- REMOTE-ACCESS VPNs typically use TLS
- SITE-TO-SITE VPNs provide SERVICE to many DEVICES within the SITES they are connecting
- REMOTE-ACCESS VPNs provide SERVICE to the ONE END DEVICE the VPN CLIENT SOFTWARE is installed on
- SITE-TO-SITE VPNs are typically used to permanently connect TWO SITES over the INTERNET
- REMOTE-ACCESS VPNs are typically used to provide ON-DEMAND ACCESS for END DEVICES that want to securely ACCESS company resources while connected to a NETWORK which is not SECURE

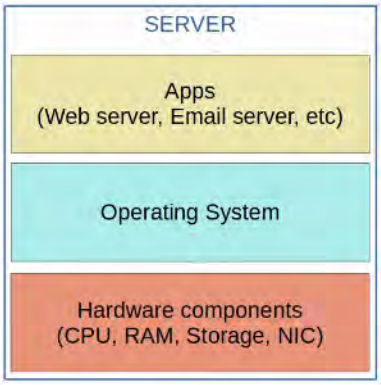
LAB COMMANDS
Create the Tunnel interface
R1(config)#int tunnel <tunnel number>
This changes the mode to the Tunnel Interface
The exit interface for the tunnel
tunnel source <interface>
IP of the Tunnel Destination Interface
tunnel destination <destination ip address>
Set the IP of the Source Tunnel Interface (from step 1)
ip address <tunnel IP> <netmask>
Configure a Default Route to the Service Provider Network
R1(config)#ip route 0.0.0.0 0.0.0.0 <next hop interface>
This will now bring the Tunnel Interface Administratively Up / Up
==========
Now you need to set up the TUNNEL ROUTERS as OSPF Neighbors for the Service Provider Network so they can share routes
R1(config)router ospf <ospf process ID>
This switches to the OSPF Router configuration mode
network <tunnel interface IP> <wildcard mask> area <area #>
Since the tunnel is a single HOST, you would use 0.0.0.0 for the Wildcard Mask
network <router gateway IP> <wildcard mask> area <area #>
Since the router gateway is also a single HOST, you would use 0.0.0.0 for the Wildcard Mask
passive-interface <router gateway IP interface>
This removes the Router Gateway from broadcasting over OSPF

54a. VIRTUALIZATION AND CLOUD: PART 1

VIRTUAL SERVERS

- Although Cisco is more known for their networking DEVICES (ROUTERS, SWITCHES, FIREWALLS), they also offer HARDWARE SERVERS such as UCS (Unified Computing System)
- The largest vendors of HARDWARE SERVERS include Dell, EMC, HPE, and IBM

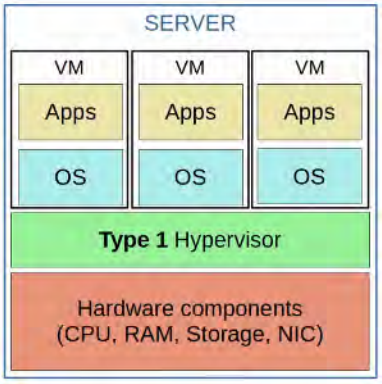
SERVERS BEFORE VIRTUALIZATION



- Before VIRTUALIZATION, there was a one-to-one relationship between a PHYSICAL SERVER and OPERATION SYSTEM
- In that OPERATING SYSTEM, apps providing SERVICES (such as a WEB SERVER, EMAIL SERVER, etc) would run
- One PHYSICAL SERVER would be used for the WEB SERVER, one for the EMAIL SERVER, one for the DATABASE SERVER, etc.
- This is inefficient for multiple reasons:
    - Each PHYSICAL SERVER is expensive and takes up space, power, etc.
    - The RESOURCES on each PHYSICAL SERVER (CPU, RAM, STORAGE, NIC) are typically under-utilized

---

VIRTUALIZATION (TYPE 1 HYPERVISOR)

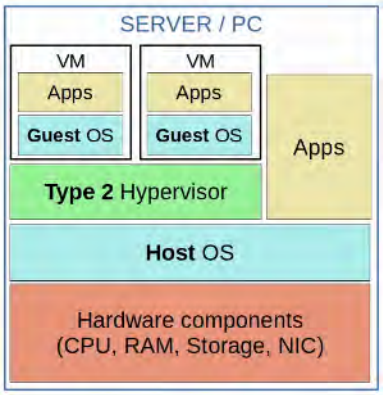


- VIRTUALIZATION allows us to break the one-to-one relationships of HARDWARE to OS, allowing multiple OS's to run on a single PHYSICAL SERVER
- Each INSTANCE is called a VM (Virtual Machine)
- A HYPERVISOR is used to manage and allocate the HARDWARE RESOURCES (CPU, RAM, etc.) to each VM
- Another name for a HYPERVISOR is VMM (Virtual Machine Monitor)
- The type of HYPERVISOR which runs directly on top of hardware is called a TYPE 1 HYPERVISOR
    - Examples include : VMware ESXi, Microsoft Hyper-V, etc.

- TYPE 1 HYPERVISORS are also called *bare-metal hypervisors* because they run directly on the hardware (metal).
    - Another term is *native hypervisor*
- This is the type of HYPERVISOR used in data center environments

---

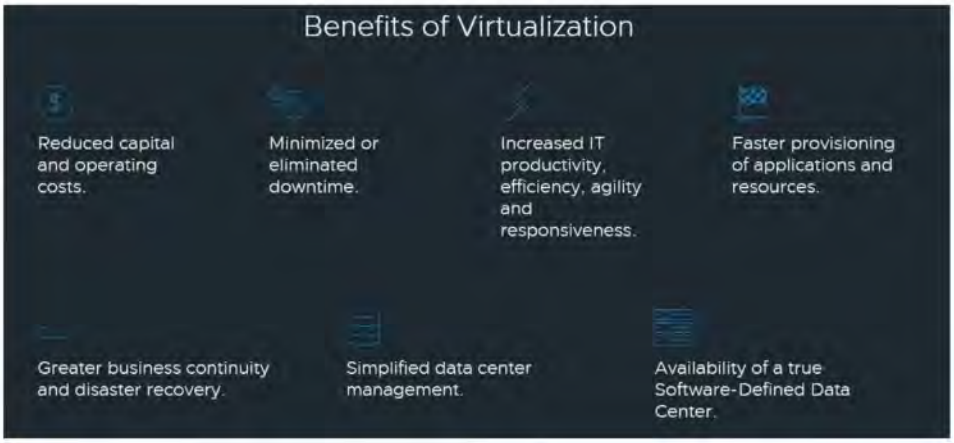
VIRTUALIZATION (TYPE 2 HYPERVISOR)



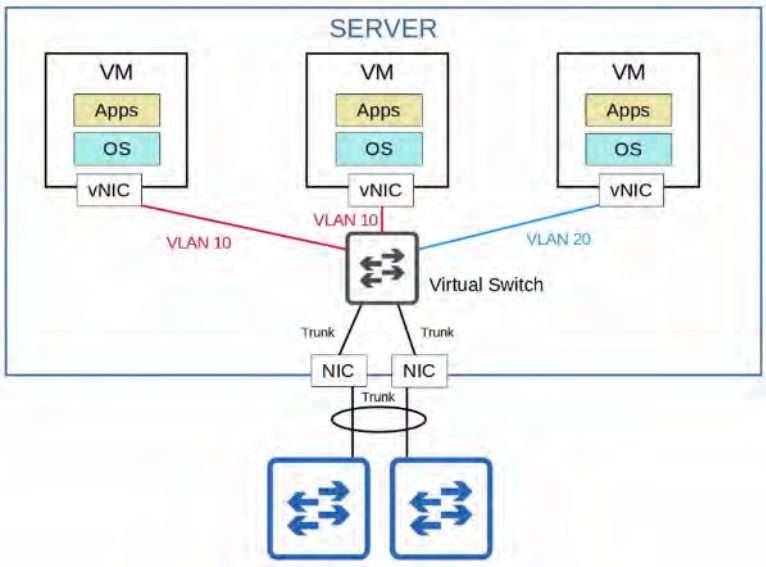
- TYPE 2 HYPERVISORS run as a program on an OS like a regular computer program
    - Examples: VMware Workstation, Oracle Virtualbox, etc
- The OS running directly on the hardware is called the HOST OS
- The OS running in a VM is called a GUEST OS
- Another name for a TYPE 2 HYPERVISOR is *hosted hypervisor*
- Although TYPE 2 HYPERVISORS are rarely used in data center environments, they are common on personal-use devices (for example, if a MAC/Linux user needs to run an app that is only supported on Windows, or vice-versa)

WHY VIRTUALIZATION?

- PARTITIONING :
    - Run multiple OS's on ONE PHYSICAL MACHINE
    - Divide system resources between VIRTUAL MACHINES
- ISOLATION :
    - Provide FAULT and SECURITY ISOLATION at the hardware level
    - Preserve performance with advanced resource controls
- ENCAPSULATION :
    - Save the entire state of a virtual machine to files
    - Move and copy virtual machines as easily as moving and copying files
- HARDWARE INDEPENDENCE :
    - Provision or migrate any virtual machine to any physical server



VIRTUAL NETWORKS



- VMs are connected to each other and the EXTERNAL NETWORK via a VIRTUAL SWITCH running on the HYPERVISOR
- Just like a regular PHYSICAL SWITCH, the vSWITCH's INTERFACES can operate as ACCESS PORTS or TRUNK PORTS and use VLANs to separate the VMs at LAYER 2
- INTERFACES on the vSWITCH connect to the PHYSICAL NIC (or NICs) of the SERVER to communicate with the EXTERNAL NETWORK

INTRO TO CLOUD COMPUTING

- Traditional IT infrastructure deployments were some combination of the following:
  - ON-PREMISES
    - All SERVERS, NETWORK DEVICES, and other infrastructure are located on company property
    - All equipment is purchased and owned by the company using it
    - The company is responsible for the necessary space, power, and cooling
  - CO-LOCATION
    - Data centers that rent out space for customers to put their infrastructure (SERVERS, NETWORK DEVICES)
    - The data center provides the space, electricity, and cooling
    - The SERVERS, NETWORK DEVICES, etc are still the responsibility of the end customer, although they are not located on the customer's premises
- CLOUD SERVICE provide an alternative that is hugely popular and is continuing to grow
  - Most people associate "CLOUD" with PUBLIC CLOUD PROVIDERS such as AWS
    - Although this is the most common USE of CLOUD SERVICES, it's not the only one

---

CLOUD SERVICES

- The American NIST (National Institute of Standards and Technology) defined cloud computing in SP (Special Publication) 800-145

> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

- To understand what the CLOUD is, let's look at the following outlined in SP 800-145:
  - FIVE ESSENTIAL CHARACTERISTICS

- THREE SERVICE MODELS
- FOUR DEPLOYMENT MODELS

THE FIVE ESSENTIAL CHARACTERISTICS OF CLOUD

- ON-DEMAND SELF-SERVICE
  - The CUSTOMER is able to use the SERVICE (or stop the SERVICE) freely (via a web portal) without direct communication to the SERVICE PROVIDER
- BROAD NETWORK ACCESS
  - The SERVICE is available through standard NETWORK connections (ie: the Internet or PRIVATE WAN) and can be access through many kinds of DEVICES
- RESOURCE POOLING
  - A POOL of RESOURCES is provided by the SERVICE PROVIDER and when a CUSTOMER requests a SERVICE (for example creates a new VM), the RESOURCES to fulfill that request are allocated from the shared POOL
- RAPID ELASTICITY
  - CUSTOMERS can quickly expand the SERVICE they use in the CLOUD (for example: add new VMs, expand STORAGE, etc) from a POOL of RESOURCES that appear to be infinite. Likewise, they can quickly reduce their SERVICES when not needed
- MEASURED SERVICE
  - The CLOUD SERVICE PROVIDER measures the CUSTOMER's usage of CLOUD RESOURCES and the CUSTOMER can measure their own use as well. CUSTOMERS are charged based on usage (for example: X Dollars per Gigabyte of STORAGE per day)

THE THREE SERVICE MODELS OF THE CLOUD

- In CLOUD COMPUTING, everything is provided on a "SERVICE" model
- For example: rather than the END USER buying a PHYSICAL SERVER, mounting it on a rack, installing the hypervisor, creating the VM, etc. the SERVICE PROVIDER offers all of this as a SERVICE
- There are a variety of SERVICES referred to as "_____ as a SERVICE" or "__aaS"
- The THREE SERVICE MODELS of CLOUD COMPUTING are:

SOFTWARE as a SERVICE (SaaS) - Example : MS Office 365

> **Software as a Service (SaaS)**. The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

PLATFORM as a SERVICE (PaaS) - Examples : AWS Lambda and Google App Engine

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

INFRASTRUCTURE as a SERVICE (Iaas) - Examples: Amazon EC2 and Google Compute Engine

Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

DEPLOYMENT MODELS

- Most people assume that “CLOUD” means PUBLIC CLOUD PROVIDERS like AWS, AZURE, and GCP
- Although “PUBLIC CLOUD” is the most common deployment model, it’s not the ONLY one
- The FOUR DEPLOYMENT MODELS of CLOUD COMPUTING are:
- PRIVATE CLOUD

Private cloud. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

- PRIVATE CLOUDS are generally only used by large enterprises
- Although the CLOUD is PRIVATE, it may be owned by a THIRD PARTY
    - For example: AWS provides PRIVATE CLOUD SERVICES for the American DoD
- PRIVATE CLOUDS may be ON or OFF PREMISES
    - Many people assume “CLOUD” and “ON-PREM” are two different things but that is not always the case
- The same kind of SERVICES offered are the same as in PUBLIC CLOUDS (SaaS, PaaS, IaaS) but the infrastructure is reserved for a SINGLE ORGANIZATION
- COMMUNITY CLOUD

Community cloud. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

- Least common CLOUD deployment
- Similar to PRIVATE CLOUD, but the INFRASTRUCTURE is reserved for use by a SPECIFIC GROUP or ORGANIZATION
- PUBLIC CLOUD

Public cloud. The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

- The most common CLOUD deployment
- Popular PUBLIC CLOUD service providers include:
    - AWS
    - MS AZURE
    - GCP (Google Cloud Platform)
    - OCI (Oracle Cloud Infrastructure)
    - IBM Cloud

    - Alibaba Cloud
- HYBRID CLOUD

> **Hybrid cloud**. The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

- Basically ANY combination of the preview THREE DEPLOYMENT TYPES
- Example: A PRIVATE CLOUD which can offload to a PUBLIC CLOUD when necessary

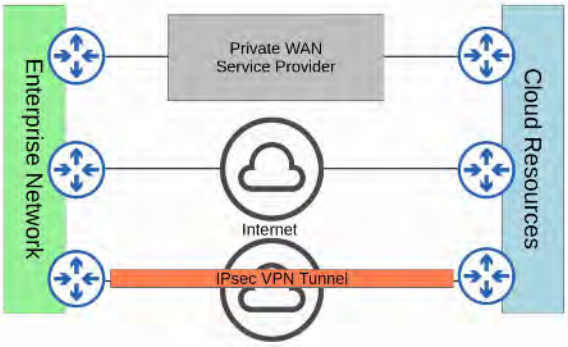
---

BENEFITS OF CLOUD COMPUTING

- COST
    - CapEx (Capital Expense) of buying HARDWARE and SOFTWARE, setting up DATA CENTERS, etc. are reduced or eliminated
- GLOBAL SCALE
    - CLOUD SERVICES can scale GLOBALLY at a rapid pace. SERVICES can be set up and offered to CUSTOMERS from a geographic location close to them
- SPEED / AGILITY
    - SERVICES are provide ON DEMAND and vast amounts of RESOURCES can be provisioned within minutes
- PRODUCTIVITY
    - CLOUD SERVICES remove the need for many time-consuming tasks such as procuring physical servers, racking them, cabling, installing and updating equipment, etc.
- RELIABILITY
    - Backups in the CLOUD are very easy to perform. Data can be mirrored at multiple sites in different geographic locations to support disaster recovery

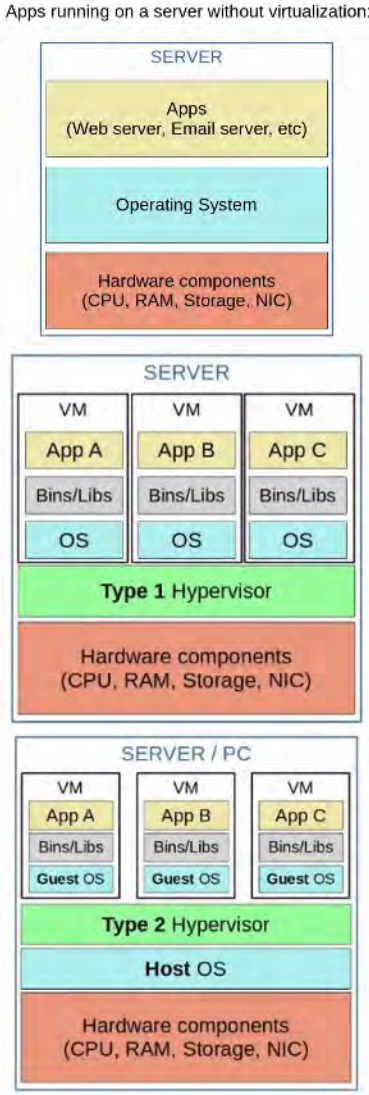
---

CONNECTION TO PUBLIC CLOUDS

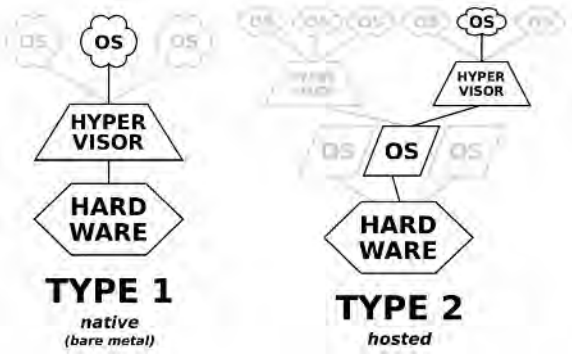


54b. VIRTUALIZATION (CONTAINERS): PART 2

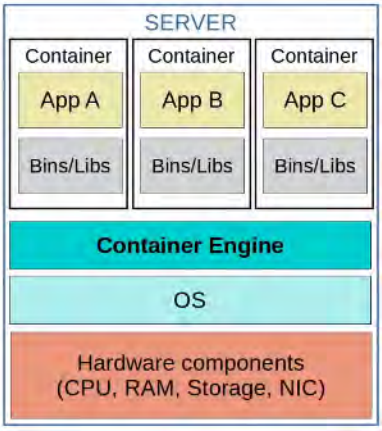
REVIEW OF VIRTUAL MACHINES (TYPE 1 and TYPE2 HYPERVISORS)



- VIRTUAL MACHINES (VMs) allow multiple OS's to run on a single PHYISCAL SERVER
- A HYPERVISOR is used to manage and allocate HARDWARE RESOURCES to each VM
    - TYPE 1 HYPERVISORS (aka NATIVE or BARE-METAL) run directly on top of HARDWARE
    - TYPE 2 HYPERVISORS (aka HOSTED) run on top of a HOST OS (ie: WINDOWS)
- TYPE 1 HYPERVISORS are widely used in DATA CENTER ENVIRONMENTS
- TYPE 2 HYPERVISORS are commonly used on personal DEVICES
    - Running a virtual network lab on your PC using Cisco Modeling Labs (CML)
- The OS in each VM can be the same or different (Windows, Linux, MacOS, etc)
- *Bins / Libs* are the SOFTWARE libraries / services needed by the Apps running in each VM
- A VM allows it's app / apps to run in an ISOLATED environment, separate from the apps in other VMs.
- VMs are easy to create, delete, move, etc.
    - A VM can be easily saved and moved between different physical SERVERS.



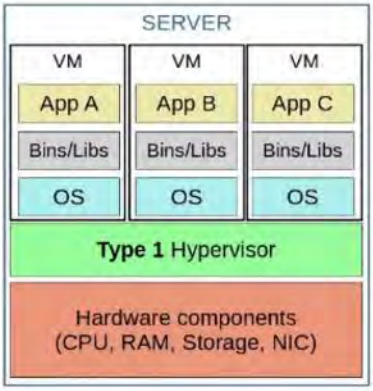
CONTAINERS

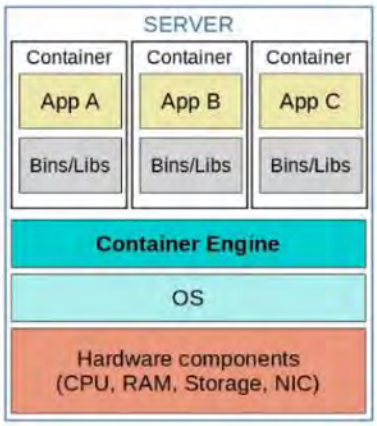


- CONTAINERS are software packages that contain an APP and all dependencies (*Bins/Libs* in the diagram) for the contained APP to run.
    - Multiple APPS can be run in a single CONTAINER, but this is not how CONTAINERS are usually used
- CONTAINERS run on a CONTAINER ENGINE (ie: DOCKER ENGINE)
    - The CONTAINER ENGINE is run on a HOST OS (usually LINUX)
- CONTAINERS are lightweight (small in size) and include only the dependencies required to run the specific APP
- A CONTAINER ORCHESTRATOR is a software platform for automating the DEPLOYMENT, MANAGEMENT, SCALING, etc of CONTAINERS
    - KUBERNETES (originally design by Google) is the most popular CONTAINER ORCHESTRATOR
    - DOCKER SWARM is DOCKER'S CONTAINER ORCHESTRATION tool
- In small numbers, MANUAL operation is possible, but large-scale systems (ie: with Microservices) can require THOUSANDS of CONTAINERS

> **Microservice Architecture** is an approach to software architecture that divides a larger solution into smaller parts (microservices).
> →Those microservices all run in containers that can be orchestrated by Kubernetes (or another platform).

VIRTUAL MACHINES vs. CONTAINERS



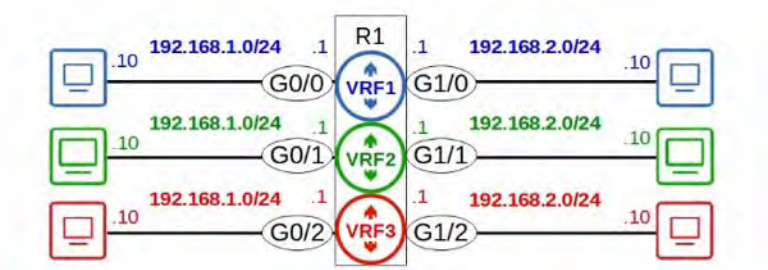


- VMs can TAKE MINUTES to boot up as each VM runs it's own OS
- CONTAINERS can boot up in milliseconds
- VMs take MORE disk space (Gigabytes)
- CONTAINERS take up VERY LITTLE disk space (Megabytes)
- VMs use MORE CPU/RAM resources (each VM must run its own OS)
- CONTAINERS use FEWER CPU/RAM resources (shared OS)
- VMs are PORTABLE and can MOVE between physical systems running the same HYPERVISOR
- CONTAINERS are MORE portable; they are SMALLER, FASTER to boot up, and DOCKER CONTAINERS can be run on nearly ANY CONTAINER SERVICE
- VMs are more isolated because each VM runs it's own OS
- CONTAINERS are less isolated because they all run on the same OS; if the OS crashes, all CONTAINERS running on it are effected

There is a major movement toward the use of containers, especially with the rise of microservices, automation, and DevOps (the combination of Software **Dev**elopment and IT **Op**eration**s**), but VMs are still widely used today.

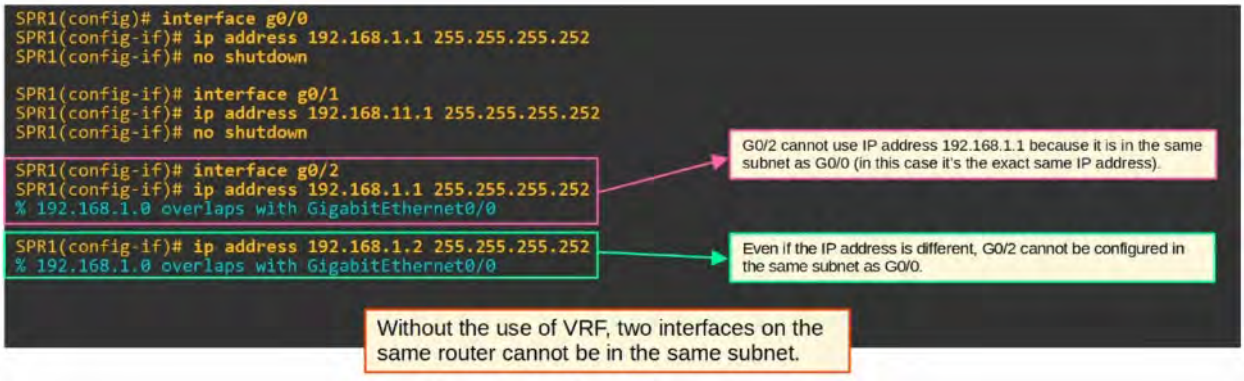
54c. VIRTUALIZATION (VRF): PART 3

INTRO TO VRF



- VIRTUAL ROUTING AND FORWARDING (VRF) is used to DIVIDE a SINGLE ROUTER into MULTIPLE VIRTUAL ROUTERS
    - Similar to how VLANs are used to divide a SINGLE SWITCH (LAN) into MULTIPLE VIRTUAL SWITCHES (VLANs)
- It does this by allowing a ROUTER to build MULTIPLE SEPARATE ROUTING TABLES
    - INTERFACES (LAYER 3 only) and ROUTERS are configured to be in a specific VRF (aka *VRF INSTANCE*)
    - ROUTER INTERFACES, SVIs and ROUTED PORTS on MULTILAYER SWITCHES can be configured in a VRF
- TRAFFIC in one VRF cannot be forwarded out of an INTERFACE in another VRF
    - As an exception, VRF LEAKING can be configured to allow traffic to pass BETWEEN VRFs
- VRF is commonly used to facilitate MPLS (Multiple Protocol Label Switching)
    - The kind of VRF we are talking about is VRF-Lite (VRF without MPLS)
- VRF is commonly used by SERVICE PROVIDERS to allow ONE DEVICE to carry traffic from MULTIPLE CUSTOMERS
    - Each CUSTOMER'S TRAFFIC is isolated from the OUTSIDE
    - CUSTOMER IP ADDRESSES can overlap without issue

VRF CONFIGURATION

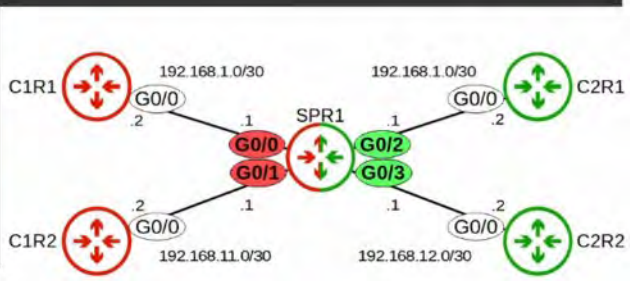


Creation and Configuration of VRFs

```
SPR1(config)# ip vrf CUSTOMER1
SPR1(config-vrf)# ip vrf CUSTOMER2
SPR1(config-vrf)# do show ip vrf
  Name                             Default RD            Interfaces
  CUSTOMER1                        <not set>
  CUSTOMER2                        <not set>

SPR1(config-vrf)# interface g0/0
SPR1(config-if)# ip vrf forwarding CUSTOMER1
% Interface GigabitEthernet0/0 IPv4 disabled and address(es) removed due to enabling VRF CUSTOMER1
SPR1(config-if)# ip address 192.168.1.1 255.255.255.252

SPR1(config-if)# interface g0/1
SPR1(config-if)# ip vrf forwarding CUSTOMER1
% Interface GigabitEthernet0/1 IPv4 disabled and address(es) removed due to enabling VRF CUSTOMER1
SPR1(config-if)# ip address 192.168.11.1 255.255.255.252

SPR1(config-if)# interface g0/2
SPR1(config-if)# ip vrf forwarding CUSTOMER2
SPR1(config-if)# ip address 192.168.1.1 255.255.255.252
SPR1(config-if)# no shutdown
SPR1(config-if)# interface g0/3
SPR1(config-if)# ip vrf forwarding CUSTOMER2
SPR1(config-if)# ip address 192.168.12.1 255.255.255.252
SPR1(config-if)# no shutdown
SPR1(config-if)# do show ip vrf
  Name                             Default RD            Interfaces
  CUSTOMER1                        <not set>             Gi0/0
                                                         Gi0/1
  CUSTOMER2                        <not set>             Gi0/2
                                                         Gi0/3
```

1. Create VRFs:
SPR1(config)# **ip vrf** *name*
2. Assign interfaces to VRFs:
SPR1(config-if)# **ip vrf forwarding** *name*

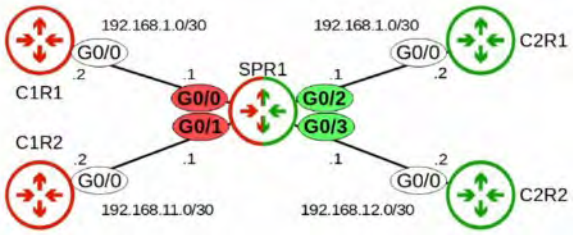
If an interface has an IP address configured, the IP address will be removed when you assign the interface to a VRF.



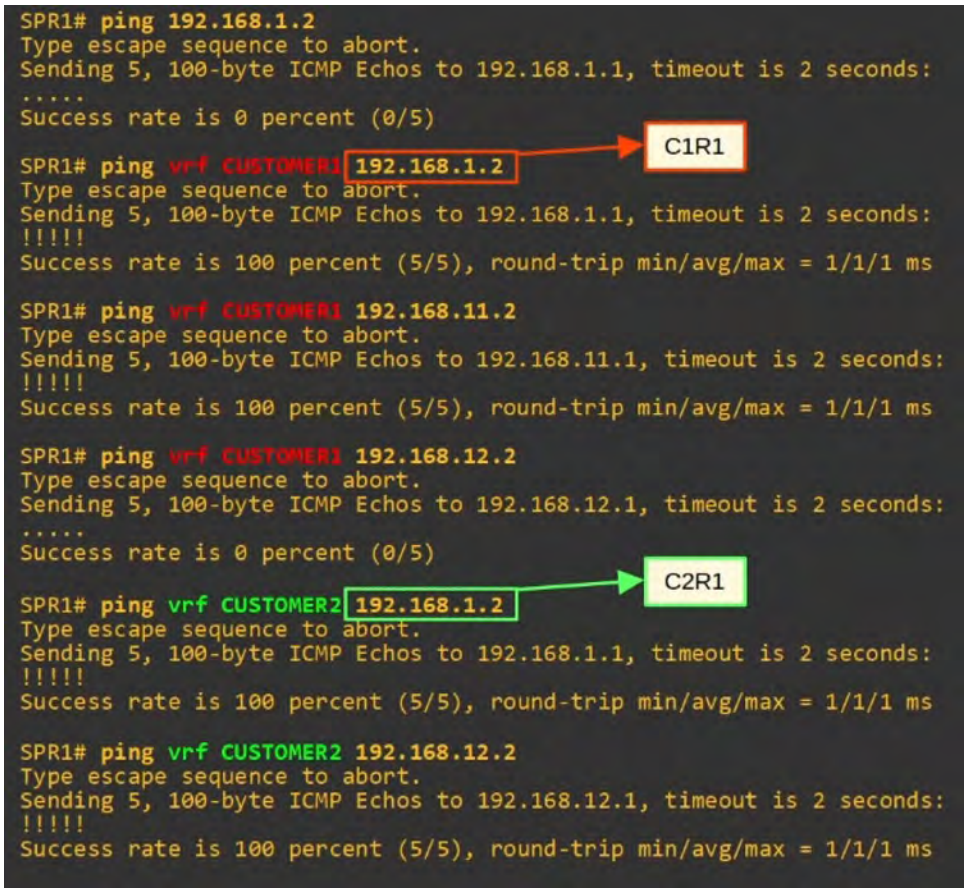
How to show ip route for VRFs

```
SPR1# 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
       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

SPR1# show ip route vrf CUSTOMER1

Routing Table: CUSTOMER1
!output omitted

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.11.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.11.0/30 is directly connected, GigabitEthernet0/1
L        192.168.11.1/32 is directly connected, GigabitEthernet0/1

SPR1# show ip route vrf CUSTOMER2

Routing Table: CUSTOMER2
!output omitted

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 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        192.168.12.0/30 is directly connected, GigabitEthernet0/3
L        192.168.12.1/32 is directly connected, GigabitEthernet0/3
```

show ip route displays the *global routing table.*
*All of SPR1's interfaces are configured in VRFs, so nothing displays here.
*You can have a mix of interfaces using and not using VRFs.



ping other VRFs

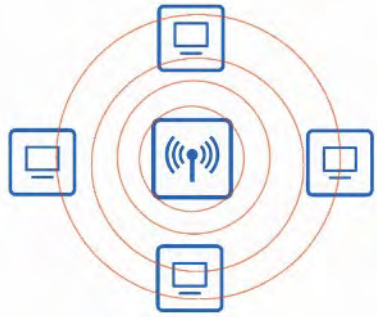

SPR1# ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
C1R1
SPR1# ping vrf CUSTOMER1 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
SPR1# ping vrf CUSTOMER1 192.168.11.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
SPR1# ping vrf CUSTOMER1 192.168.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
C2R1
SPR1# ping vrf CUSTOMER2 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
SPR1# ping vrf CUSTOMER2 192.168.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

55. WIRELESS FUNDAMENTALS

- Although we will briefly look at other types of WIRELESS NETWORKS, in this section of the course we will be focusing on WIRELESS LANs using WI-FI
- The STANDARDS we use for WIRELESS LANs are defined in IEEE 802.11
- The term WI-FI is a trademark of the WI-FI ALLIANCE, not directly connected to the IEEE
- The WI-FI ALLIANCE tests and certifies equipment for 802.11 standards compliance
- However, WI-FI has become the common term that people use to refer to 802.11 WIRELESS LANs and that term will be used through the course videos

WIRELESS NETWORKS

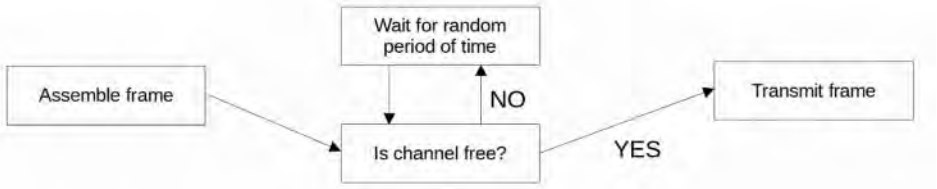
- WIRELESS NETWORKS have some issues that we need to deal with



1. ALL DEVICES within range receive ALL FRAMES, like DEVICES connected to an ETHERNET HUB

- Privacy of DATA within the LAN is a greater concern
- CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) is used to facilitate HALF-DUPLEX communications
- CSMA / CD is used in WIRED NETWORKS to detect and recover from COLLISIONS
- CSMA / CA is used in WIRELESS NETWORKS to avoid COLLISIONS
- When using CSMA / CA, a DEVICE will wait for other DEVICES to STOP TRANSMITTING before it TRANSMITS DATA itself.

5

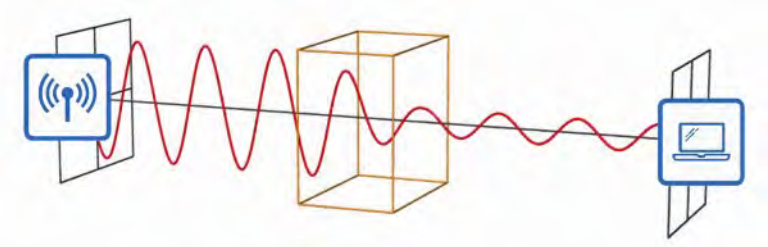


2. WIRELESS COMMUNICATIONS are regulated by various INTERNATIONAL and NATIONAL bodies
3. WIRELESS SIGNAL COVERAGE AREA must be considered

- Signal Range
- Signal ABSORPTION, REFLECTION, REFRACTION, DIFFRACTION, and SCATTERING

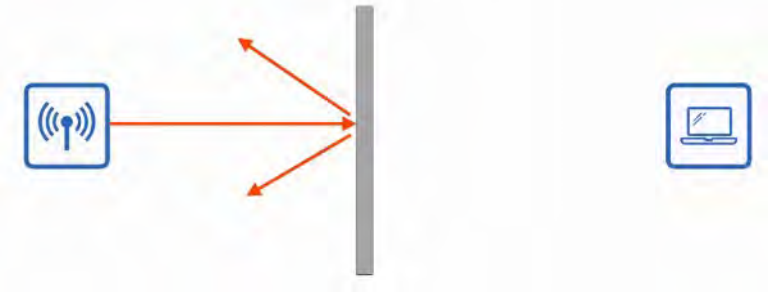
SIGNAL ABSORPTION

- ABSOPTION happens when a WIRELESS SIGNAL PASSES THROUGH a material and is converted into HEAT, weakening the SIGNAL



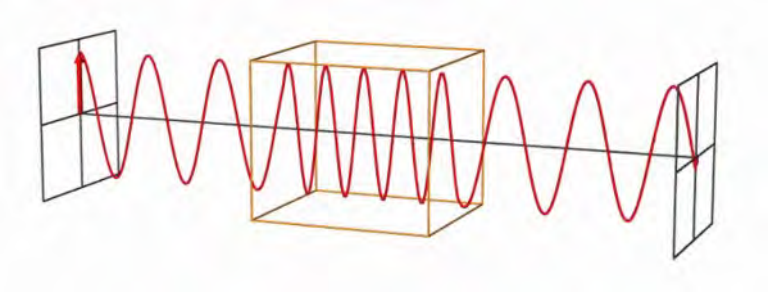
SIGNAL REFLECTION

- REFLECTION happens when a SIGNAL BOUNCES off a material (like metal)
    - This is why WI-FI reception is usually POOR in elevators. The SIGNAL bounces off the metal and very little penetrates into the elevator



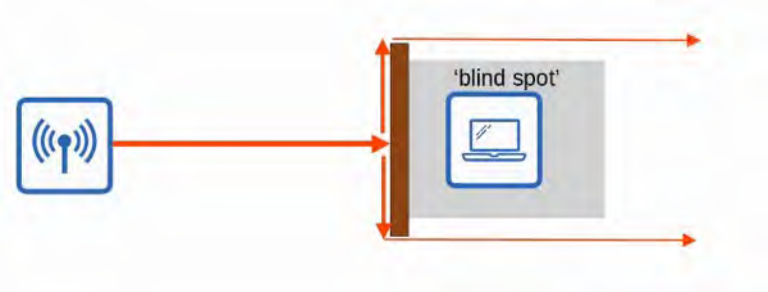
SIGNAL REFRACTION

- REFRACTION happens when a WAVE is BENT when entering a medium where the SIGNAL travels at a different speed
    - For example, glass and water can refract waves



SIGNAL DIFFRACTION

- DIFFRACTION happens when a WAVE encounters an OBSTACLE and travels AROUND it
    - This can result in “BLIND SPOTS” behind the obstacle



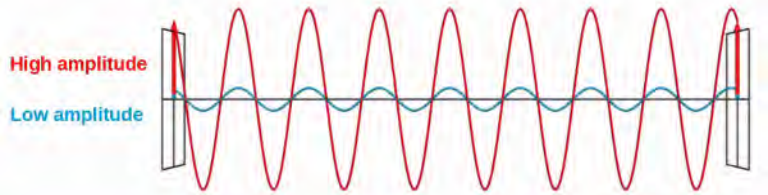
SIGNAL SCATTERING

- SCATTERING happens when a material causes a SIGNAL to SCATTER in all directions
    - Dust, smog, uneven surfaces, etc. can cause scattering

4. Other DEVICES using the SAME CHANNELS can cause INTERFERENCE
- For example, a WIRELESS LAN in your neighbor's house / apartment

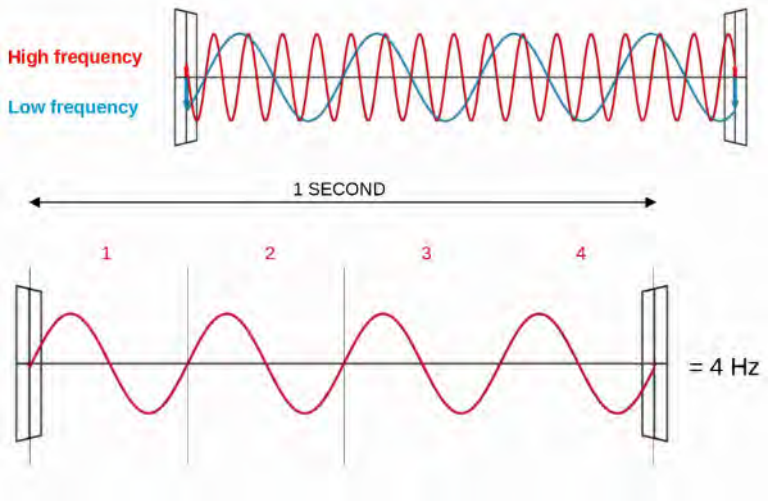
RADIO FREQUENCY (RF)

- To send WIRELESS SIGNALS, the SENDER applies an ALTERNATING CURRENT to an antenna
    - This creates ELECTROMAGNETIC WAVES which propagate out as WAVES
- ELECTROMAGENETIC WAVES can be measured in multiple ways - for example AMPLITUDE and FREQUENCY
- AMPLITUDE is the MAXIMUM STRENGTH of the ELECTRIC and MAGNETIC FIELDS

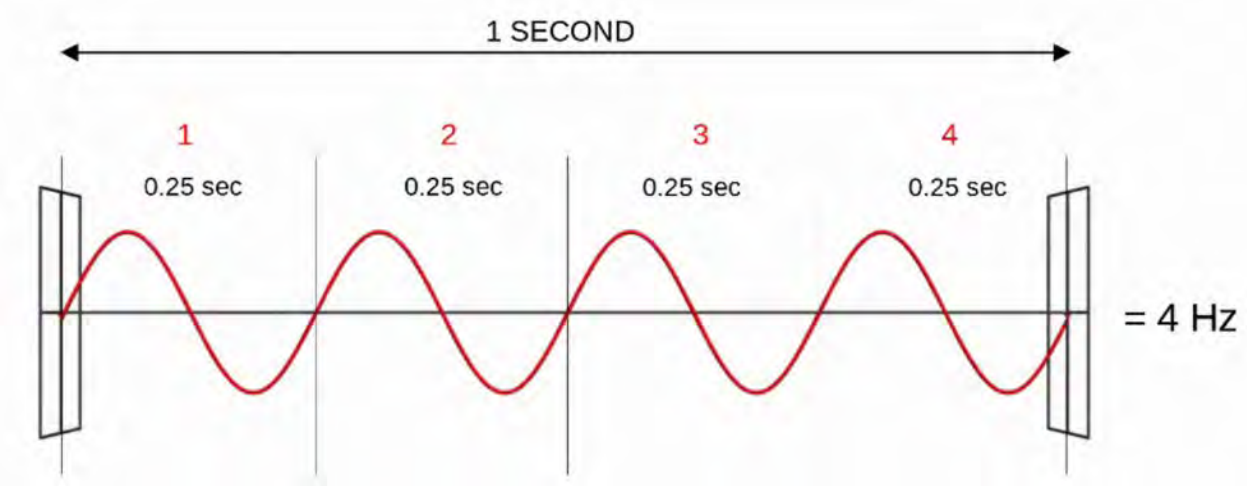


- FREQUENCY measures the number of UP / DOWN CYCLES per a GIVEN UNIT of TIME
- The most COMMON measurement of FREQUENCE is HERTZ
    - Hz (HERTZ) = cycles per second
    - kHz (KILOHERZ) = 1,000 cycles per second
    - MHz (MEGAHERZ) = 1,000,000 cycles per second
    - GHz (GIGAHERTZ) = 1,000,000,000 cycles per second
    - THz (TERAHERTZ) = 1,000,000,000,000 cycles per second

4 CYCLES per 1 SECOND = 4 HERTZ



- Another important term is PERIOD, the amount of TIME of ONE CYCLE
    - If the FREQUENCY is 4 Hz, the PERIOD is 0.25 SECONDS



- The VISIBLE FREQUENCY RANGE is ~400 THz to 790 THz
- The RADIO FREQUENCY RANGE is 30 Hz to 300 GHz and is used for many purposes.

Ultra high frequency	UHF	9	300–3,000 MHz 1–0.1 m	Television broadcasts, microwave oven, microwave devices/communications, radio astronomy, mobile phones, wireless LAN, Bluetooth, ZigBee, GPS and two-way radios such as land mobile, FRS and GMRS radios, amateur radio, satellite radio, Remote control Systems, ADSB.
Super high frequency	SHF	10	3–30 GHz 100–10 mm	Radio astronomy, microwave devices/communications, wireless LAN, DSRC, most modern radars, communications satellites, cable and satellite television broadcasting, DBS, amateur radio, satellite radio.

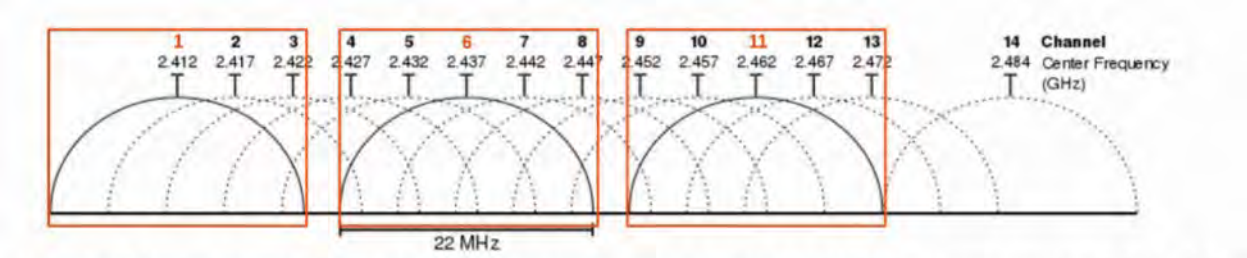
RADIO FREQUENCY BANDS

- WI-FI uses TWO MAIN BANDS (FREQUENCY RANGES)
- 2.4 GHz band
  - Range is 2.400 - 2.4835 GHz
- 5 GHz band
  - Range is 5.150 - 5.825 GHz
  - Divided into FOUR SMALLER BANDS:
    - 5.150 - 5.250 GHz
    - 5.250 - 5.350 GHz
    - 5.470 - 5.725 GHz
    - 5.725 - 5.825 GHz
- The 2.4 GHz band typically provides FURTHER REACH in open space and BETTER PENETRATION of obstacles such as walls.
  - HOWEVER, more DEVICES tend to use the 2.4 GHz BAND so INTERFERENCE can be a BIGGER PROBLEM compared to 5GHz

** WI-FI 6 (802.11ax) has EXPANDED the spectrum range to include a band in the 6 GHz RANGE

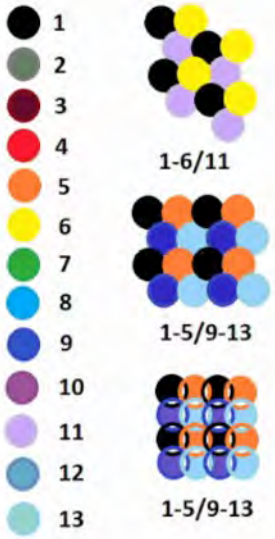
CHANNELS

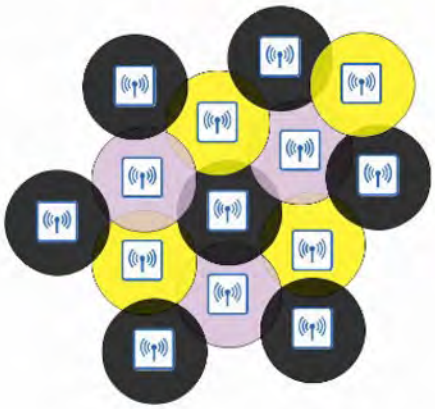
- Each BAND is divided up into MULTIPLE “CHANNELS”
  - DEVICES are configured to TRANSMIT and RECEIVE traffic on one (or more) of these CHANNELS
- The 2.4 GHz BAND is divided into several CHANNELS, each with a 22 MHz RANGE
- In a SMALL WIRELESS LAN with only a single ACCESS POINT (AP), you can use ANY channel
- However, in larger WLANs with multiple APs, it’s important that adjacent APs don’t use OVERLAPPING CHANNELS. This helps avoid INTERFERENCE
- In the 2.4 GHz BAND, it is recommended to use CHANNELS 1, 6 and 11



Outside of North America you could use other combinations, but for the CCNA exam remember **1, 6, and 11**.

- The 5 GHz BAND consists of NON-OVERLAPPING channels so it's much EASIER to avoid INTERFERENCE between adjacent APs
- Using CHANNELS 1, 6, 11, you can place APs in a "HONEYCOMB" pattern to provide COMPLETE coverage of an area without INTERFERENCE between CHANNELS





WI-FI STANDARDS (802.11)

Standard	Frequencies	Max Data Rate (theoretical)	Alternate Name
802.11	2.4 GHz	2 Mbps	
802.11b	2.4 GHz	11 Mbps	
802.11a	5 GHz	54 Mbps	
802.11g	2.4 GHz	54 Mbps	
802.11n	2.4 / 5 GHz	600 Mbps	'Wi-Fi 4'
802.11ac	5 GHz	6.93 Gbps	'Wi-Fi 5'
802.11ax	2.4 / 5 / 6 GHz	4*802.11ac	Wi-Fi 6'

SERVICE SETS

- 802.11 defines different kinds of SERVICE SETS which are groups of WIRELESS NETWORK DEVICES
- There are THREE MAIN TYPES:

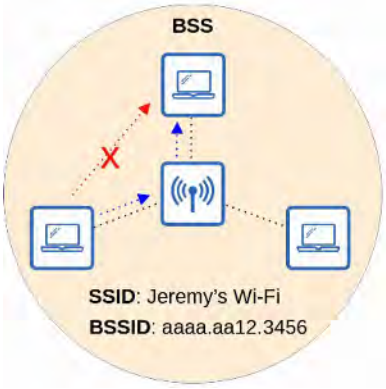
    - INDEPENDENT
    - INFRASTRUCTURE
    - MESH
- ALL DEVICES in a SERVICE SET share the same SSID (Service Set Identifier)
- The SSID is a HUMAN-READABLE NAME which identifies the SERVICE SET
- The SSID does NOT have to be UNQUE

SERVICE SETS : IBSS

- An IBSS (INDEPENDENT BASIC SERVICE SET) is a WIRELESS NETWORK in which TWO or MORE WIRELESS DEVICES connect directly without using an AP (ACCESS POINT)
- Also called an AD HOC NETWORK
- Can be used for FILE TRANSFER (ie: AirDrop)
- Not scalable beyond a few DEVICES

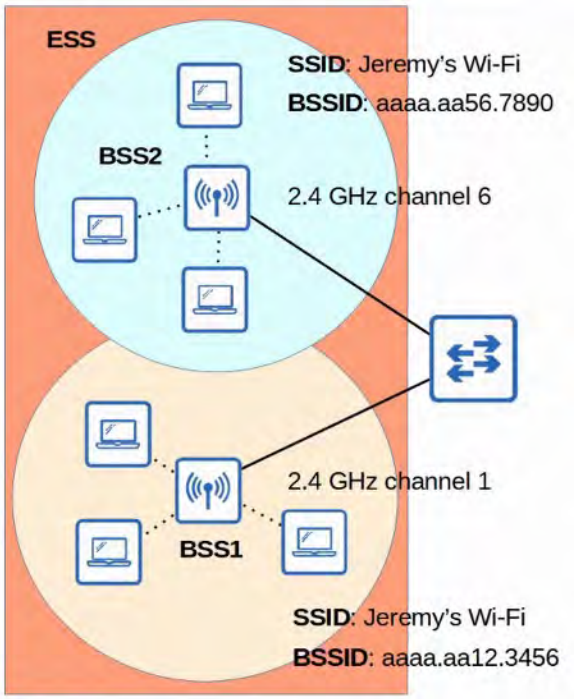
SERVICE SETS : BSS

- A BSS (BASIC SERVICE SET) is a kind of infrastructure SERVICE SET in which CLIENTS connect to each other via an AP (ACCESS POINT) but not DIRECTLY to each other
- A BSSID (BASIC SERVICE SET ID) is used to uniquely identify the AP
    - Other APs can use the SAME SSID but NOT THE SAME BSSID
    - The BSSID is the MAC ADDRESS of the APs RADIO
- WIRELESS DEVICES request to *associate* with the BSS
- WIRELESS DEVICES that have associated with the BSS are called “CLIENTS” or “STATIONS”
- The AREA around an AP where its SIGNAL is usable is called a BSA (BASIC SERVICE AREA)



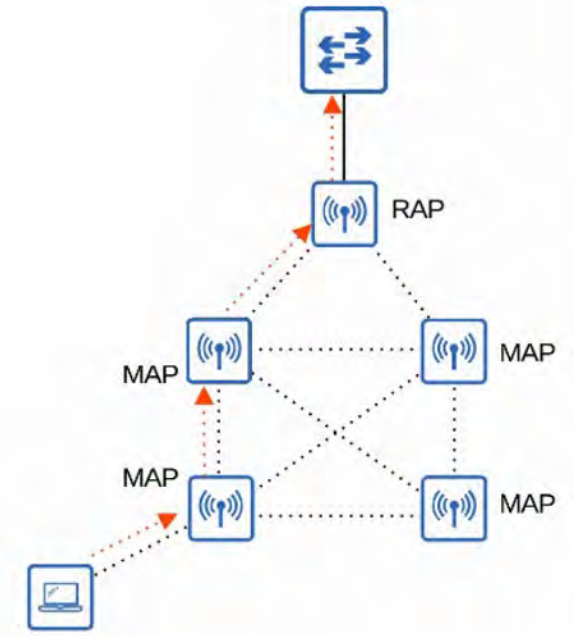
SERVICE SETS: ESS

- To create LARGER WIRELESS LANS beyond the range of a SINGLE AP, we use an ESS (EXTENDED SERVICE SET)
- APs with their own BSSs are connected by a WIRED NETWORK
    - Each BSS uses the SAME SSID
    - Each BSS has a UNIQUE BSSID
    - Each BSS uses a DIFFERENT channel to avoid INTERFERENCE
- CLIENTS can pass between APs without having to RECONNECT, providing a SEAMLESS WI-FI experience when moving between APs
    - This is called ROAMING
- The BSAs should overlap about 10-15%



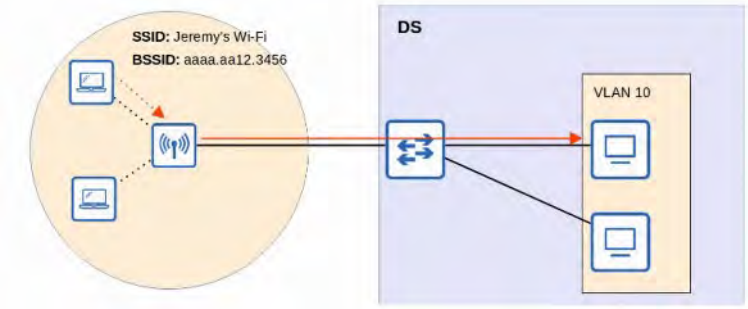
SERVICE SETS: MBSS

- An MBSS (MESH BASIC SERVICE SET) can be used in situations where it's difficult to run an ETHERNET connection to every AP
- MESH APs use TWO RADIOS:
    - ONE provides BSS to WIRELESS CLIENTS
    - ONE forms a "BACKHAUL NETWORK" which is used to BRIDGE traffic from AP to AP
- At least ONE AP is connected to the WIRED NETWORK and it is called the RAP (ROOT ACCESS POINT)
- The OTHER APs are called MAPs (MESH ACCESS POINTS)
- A PROTOCOL is used to determine the BEST PATH through the MESH (similar to how DYNAMIC ROUTING PROTOCOLS are used to determine the BEST PATH to a DESTINATION)

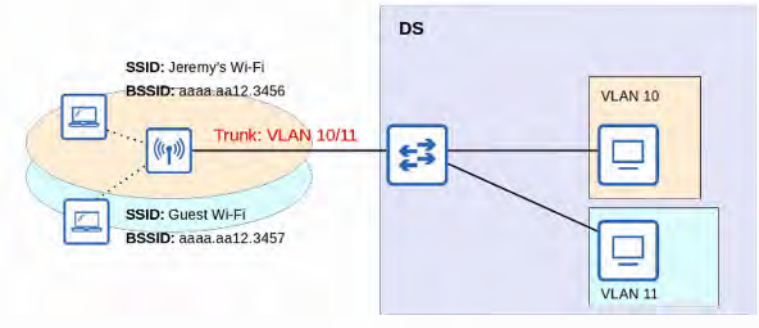


DISTRIBUTION SYSTEM

- Most WIRELESS NETWORKS are not STANDALONE NETWORKS
  - Rather, they are a way for WIRELESS CLIENTS to connect to the WIRED NETWORK INFRASTRUCTURE
- In 802.11, the UPSTREAM WIRED NETWORK is called the DS (DISTRIBUTION SYSTEM)
- Each WIRELESS BSS or ESS is mapped to a VLAN in the WIRED NETWORK

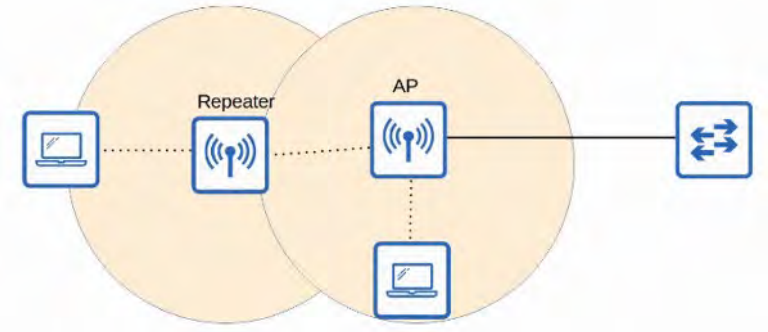


- It is possible for an AP to provide MULTIPLE WIRELESS LANs, each with a unique SSID
- Each WLAN is mapped to a separate VLAN and connected to the WIRED NETWORK via a TRUNK
- Each WLAN uses a UNIQUE BSSID, usually by INCREMENTING the LAST digit of the BBSID by one



ADDITIONAL AP OPERATIONAL MODES

- APs can operate in ADDITIONAL MODES beyond the ones we've introduced so far
- An AP in REPEATER MODE can be used to EXTEND the RANGE of a BSS
- The REPEATER will re-transmit ANY SIGNAL it receives from the AP
  - A REPEATER with a SINGLE RADIO must operate on the SAME CHANNEL as the AP, but this can drastically reduce the overall THROUGHPUT on the CHANNEL
  - A REPEATER with TWO RADIOS can receive on ONE CHANNEL and then retransmit on ANOTHER CHANNEL



- A WORKGROUP BRIDGE (WGB) operates as a WIRELESS CLIENT of another AP and can be used to CONNECT WIRED DEVICES to the WIRELESS NETWORK
- In the example below, PC1 does NOT have WIRELESS CAPABILITIES, and also DOES NOT have ACCESS to WIRED CONNECTIONS to SW1
- PC1 has a WIRED CONNECTION to the WGB, which has a WIRELESS CONNECTION to the AP

There are two kinds of WGBs:
Universal WGB (uWGB) is an 802.11 standard that allows one device to be bridged to the wireless network.

WGB is a Cisco-proprietary version of the 802.11 standard that allows multiple wired clients to be bridged to the wireless network.

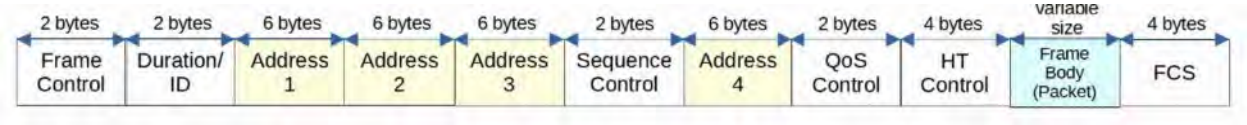
- AN OUTDOOR BRIDGE can be used to connect NETWORKS over LONG DISTANCES without a PHYSICAL CABLE connecting them
- The APs will use SPECIALIZED ANTENNAS that focus most of the SIGNAL POWER in one direction, which allows the WIRELESS CONNECTION to be made over LONGER DISTANCES than normally possible
- The CONNECTION can be POINT-TO-POINT as in the diagram below, or POINT-TO-MULTIPOINT in which MULTIPLE SITES connect to on CENTRAL SITE

REVIEW

- Wireless LANs are defined in 802.11.
- Operate in half duplex using CSMA/CA
- Wireless signals can be affected by **absorption, reflection, refraction, diffraction, and scattering.**
- Various aspects of waves can be measured, such as **amplitude, frequency,** and **period.**
- Frequency is measured in **hertz** (Hz).
- Wireless LANs use two frequency ranges: the **2.4 GHz** band and **5 GHz** band.
  → Wi-Fi 6 (802.11ax) can use the **6 GHz** range too.
- Bands are divided into channels.
- 5 GHz band consists of non-overlapping channels.
- 2.4 GHz band channels overlap. To avoid overlapping, use channels 1, 6, and 11 (in North America).
- 802.11 standards (802.11b, 802.11a, etc) and their frequencies/theoretical max data rates.
- Service sets are groups of wireless devices. Three types:
  → Independent (**IBSS**, also called **ad hoc**)
  → Infrastructure (**BSS**, **ESS**)
  *passing between APs in an ESS is called **roaming**.
  → Mesh (**MBSS**)
- Service sets are identified by an **SSID** (non-unique, human-readable) and **BSSID** (unique, MAC address of AP).
- The area around an AP where its signal is usable is called a **BSA**.
- The upstream wired network is called the **DS**.
- When multiple WLANs are used, each is mapped to a separate VLAN on the wired network.
- APs can also operate as a **repeater**, **workgroup bridge**, or **outdoor bridge**.

56. WIRELESS ARCHITECTURES

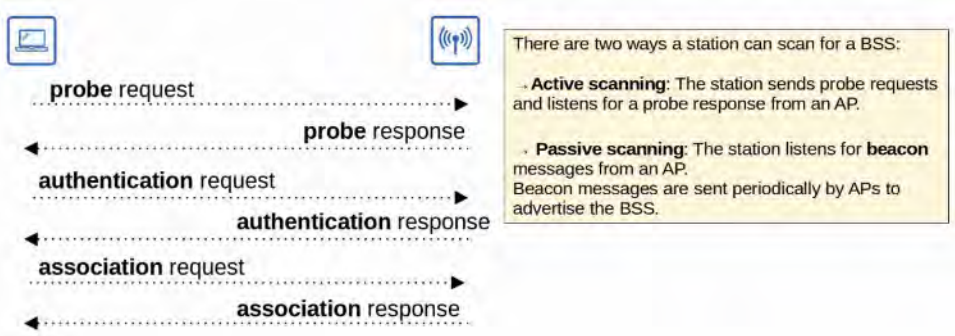
802.11 MESSAGE / FRAME FORMAT



- 802.11 FRAMES have a different format than 802.3 ETHERNET FRAMES
- For the CCNA, you don't have to learn it in as much detail as the ETHERNET and IP HEADERS
- Depending on the 802.11 VERSION and the MESSAGE TYPE, some of the fields might not be present in the FRAME
    - For example: Not ALL messages use all 4 ADDRESS FIELDS
- FRAME CONTROL
    - Provides information such as MESSAGE TYPE and SUBTYPE
    - Indicates if the FRAME is a MANAGEMENT frame
- DURATION / ID
    - Depending on the MESSAGE TYPE, this field can indicate:
        - The TIME (in microseconds) the CHANNEL will be dedicated to transmission of the FRAME
        - Identifier for the ASSOCIATION (the connection)
- ADDRESSES
    - Up to FOUR ADDRESSES can be present in an 802.11 FRAME.
    - Which ADDRESSES are present, and their ORDER, depends on the MESSAGE TYPE
        - DESTINATION ADDRESS (DA) : Final RECIPIENT of the FRAME
        - SOURCE ADDRESS (SA) : Original SENDER of the FRAME
        - RECEIVER ADDRESS (RA) : Immediate RECIPIENT of the FRAME
        - TRANSMITTER ADDRESS (TA) : Immediate SENDER of the FRAME
- SEQUENCE CONTROL
    - Used to reassemble FRAGMENTS and eliminate DUPLICATE FRAMES
- QoS CONTROL
    - Used in QoS to PRIORITIZE certain traffic
- HT (High Throughput) CONTROL
    - Added in 802.11n to ENABLE High Throughput operations
    - 802.11n is also known as "HIGH THROUGHPUT" (HT) WI-FI
    - 802.11ac is also know as "VERY HIGH THROUGHPUT" (VHT) WI-FI
- FCS (FRAME CHECK SEQUENCE)
    - Same as in an ETHERNET FRAME, used to check for errors

802.11 ASSOCIATION PROCESS

- ACCESS POINTS bridge traffic between WIRELESS STATIONS and other DEVICES
- For a STATION to send traffic through the AP, it must be associated with the AP
- There are THREE 802.11 CONNECTION STATES:
    - NOT AUTHENTICATED, NOT ASSOCIATED
    - AUTHENTICATED, NOT ASSOCIATED
    - AUTHENTICATED and ASSOCIATED
- The STATION must be AUTHENTICATED and ASSOCIATED with the AP to send traffic through it

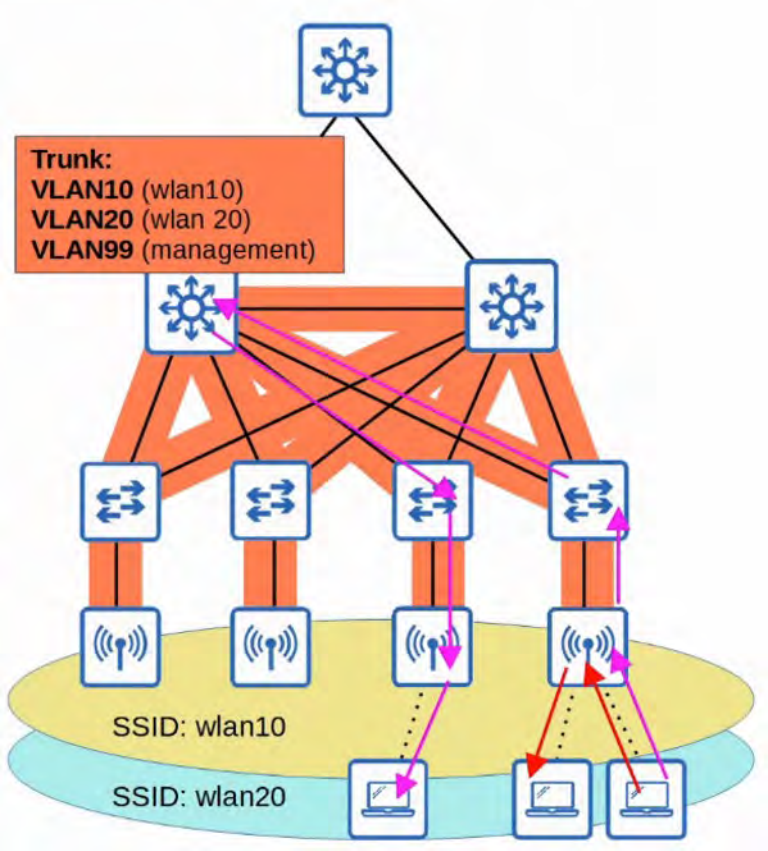


802.11 MESSAGE TYPES

- There are THREE 802.11 MESSAGE TYPES
  - MANAGEMENT
  - CONTROL
  - DATA
- MANAGEMENT
  - Used to manage the BSS
    - BEACON
    - PROBE REQUEST / PROBE RESPONSE
    - AUTHENTICATION
    - ASSOCIATION REQUEST / ASSOCIATION RESPONSE
- CONTROL
  - Used to control access to the medium (RADIO FREQUENCY)
  - Assists with delivery of MANAGEMENT and DATA FRAMES
    - RTS (REQUEST TO SEND)
    - CTS (CLEAR TO SEND)
    - ACK
- DATA
  - Used to send actual DATA PACKETS

AUTONOMOUS APs

- AUTONOMOUS APs are self-contained SYSTEMS that do NOT RELY on a WLC
- AUTONOMOUS APs are configured individually
  - Can be configured by CONSOLE cable (CLI)
  - Can be configured by TELNET (CLI)
  - Can be configured by HTTP / HTTPS Web connection (GUI)
  - An IP ADDRESS for REMOTE MANAGEMENT should be configured
  - The RF PARAMETERS must be manually configured (Transmit Power, Channel, etc)
  - SECURITY POLICIES are handled individually by each AP
  - QoS RULES etc. are configured individually by each AP
- There is NO CENTRAL MONITORING or MANAGEMENT of APs

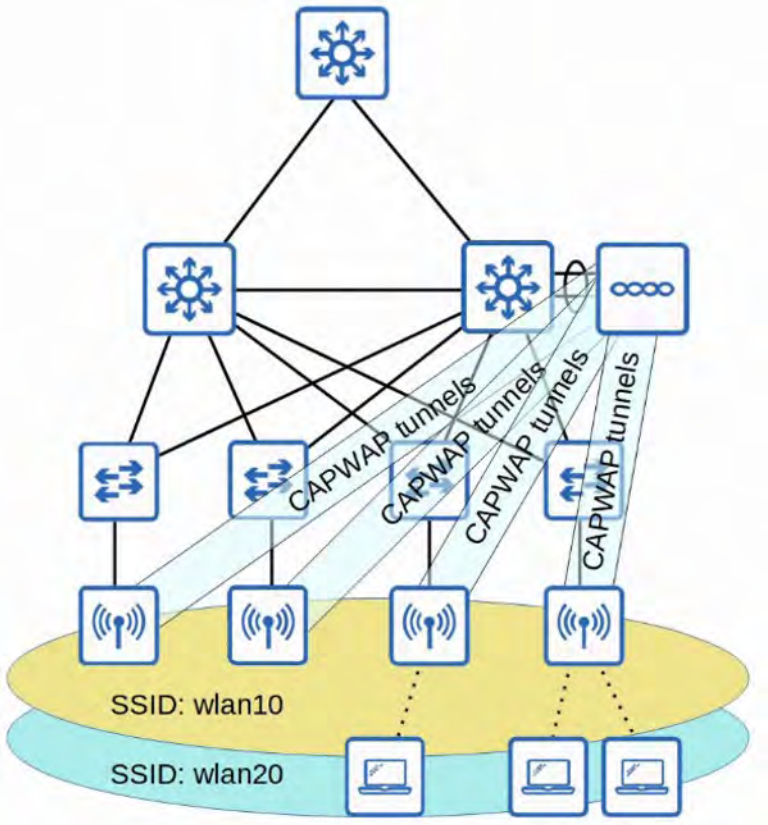


- AUTONOMOUS APs connect to the WIRED NETWORK with a TRUNK link
- DATA traffic from WIRELESS CLIENTS have a very direct PATH to the WIRED NETWORK or to other WIRELESS CLIENTS associated with the same AP
- Each VLAN has to STRETCH across the entire NETWORK. This is considered BAD practice
    - Large Broadcast Domains
    - Spanning Tree will disable links
    - Adding / Deleting VLANs is VERY labor-intensive
- AUTONOMOUS APs can be used in SMALL NETWORKS but they are not viable in MEDIUM to LARGE NETWORKS
    - LARGE NETWORKS can have thousands of APs
- AUTONOMOUS APs can also function in the modes covered in the previous video:
    - REPEATER
    - OUTDOOR BRIDGE
    - WORKGROUP BRIDGE

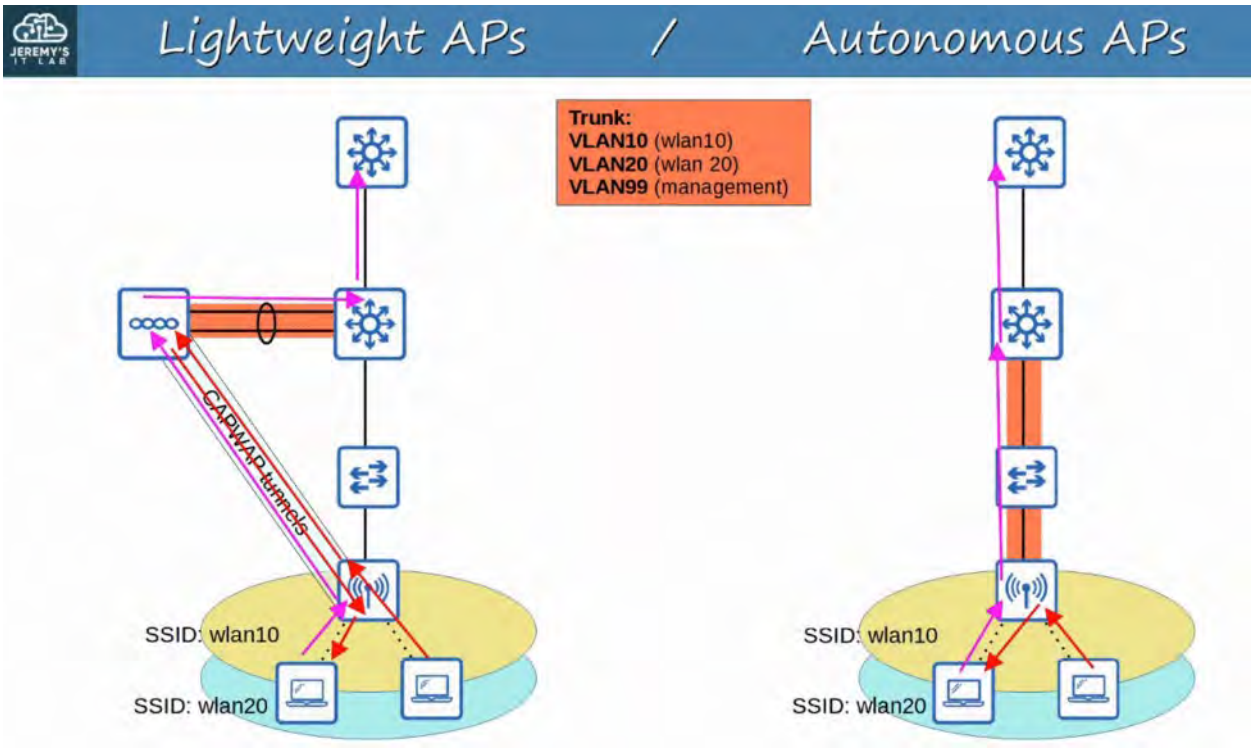
LIGHTWEIGHT APs

- The functions of an AP can be split between the AP and a WIRELESS LAN CONTROLLER (WLC)
- The is what is called SPLIT-MAC ARCHITECTURE
- LIGHTWEIGHT APs handle **“real-time”** operations like:
    - TRANSMITTING / RECEIVING RF TRAFFIC
    - ENCRYPTION / DECRYPTION OF TRAFFIC
    - SENDING OUT BEACONS / PROBES
    - PACKET PRIORITIZATION
    - Etc…
- WLC Functions (not time dependent)
    - RF MANAGEMENT
    - SECURITY / QoS MANAGEMENT

    - CLIENT AUTHENTICATION
    - CLIENT ASSOCIATION / ROAMING MANAGEMENT
    - RESOURCE ALLOCATION
    - Etc…
- The WLC is also used to centrally configured the lightweight APs
- The WLC can be located in the same SUBNET / VLAN as the lightweight APs it manages OR in a different SUBNET / VLAN
- The WLC and the lightweight APs AUTHENTICATE each other using DIGITAL CERTIFICATES installed on each DEVICE (X.509 STANDARD CERTIFICATES)
    - This ensures that only AUTHORIZED APs can join the NETWORK



- THE WLC and lightweight APs use a PROTOCOL called CAPWAP (CONTROL AND PROVISIONING OF WIRELESS ACCESS POINTS) to communicate
    - Based on an older PROTOCOL called LWAPP (LIGHTWEIGHT ACCESS POINT PROTOCOL)
- TWO TUNNELS are created between each AP and the WLC :
    - CONTROL TUNNEL (UDP Port 5246)
        - This TUNNEL is used to configure the APs and control and manage operations
        - All traffic in this TUNNEL is ENCRYPTED, by default
    - DATA TUNNEL (UDP Port 5247)
        - All traffic from WIRELESS CLIENTS is sent through this TUNNEL to the WLC
        - IT DOES NOT GO DIRECTLY TO THE WIRED NETWORK !
- Traffic in this TUNNEL is not ENCRYPTED by default but you can configure it to be ENCRYPTED with DTLS (DATAGRAM TRANSPORT LAYER SECURITY)
- Because ALL traffic from WIRELSS CLIENTS is TUNNELED to the WLC with CAPWAP, APs connect to the SWITCH ACCESS PORTS - NOT TRUNK PORTS

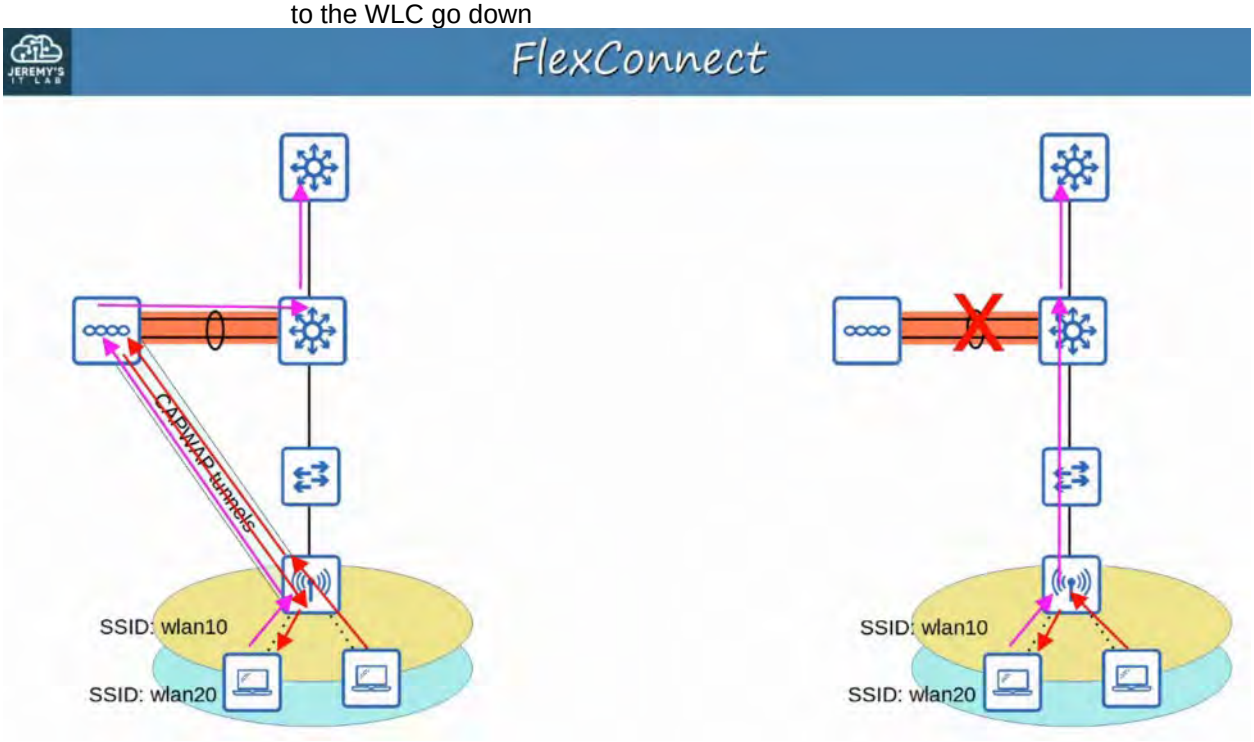


*** (Not necessary to MEMORIZE for CCNA) ***
There are some KEY BENEFITS to SPLIT-MAC ARCHITECTURE

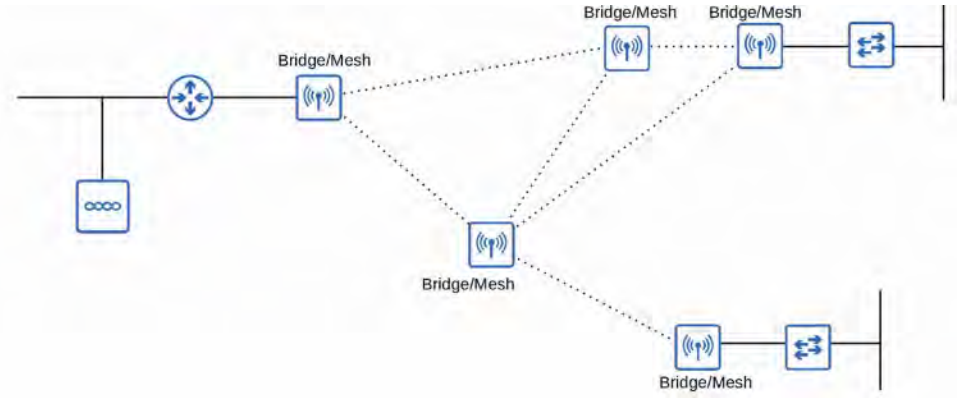
- SCALABILITY
  - With a WLC (or multiple) it's SIMPLER to build and support a NETWORK with thousands of APs
- DYNAMIC CHANNEL ASSIGNMENT
  - The WLC can automatically select which channel each AP should use
- TRANSMIT POWER OPTIMIZATION
  - The WLC can automatically set the appropriate transmit power for each AP
- SELF-HEALING WIRELESS COVERAGE
  - When an AP stops functioning, the WLC can increase the transmit power of nearby APs to avoid coverage holes
- SEAMLESS ROAMING
  - CLIENTS can roam between APs with no noticeable delay
- CLIENT LOAD BALANCING
  - If a CLIENT is in range of TWO APs, the WLC can associate the CLIENT with the least-used AP, to balance the load among APs
- SECURITY / QoS MANAGEMENT
  - Central management of SECURITY and QoS policies ensures consistency across the NETWORK

- LIGHTWEIGHT APs can be configured to operate in VARIOUS MODES:
  - LOCAL
    - This is the DEFAULT mode where the AP offers a BSS (more multiple BSSs) for CLIENTS to associate with
  - FLEXCONNECT
    - Like a LIGHTWEIGHT AP in LOCAL mode, it offers ONE or MORE BSSs for CLIENTS to associate with

- HOWEVER, FLEXCONNECT allows the AP to locally SWITCH traffic between the WIRED (TRUNK) and WIRELESS NETWORKS (ACCESS) if the TUNNELS to the WLC go down

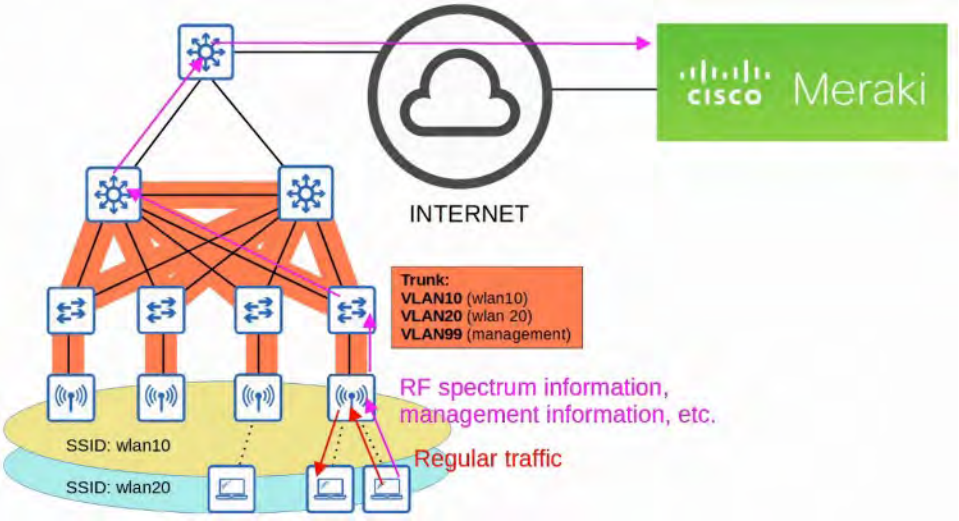


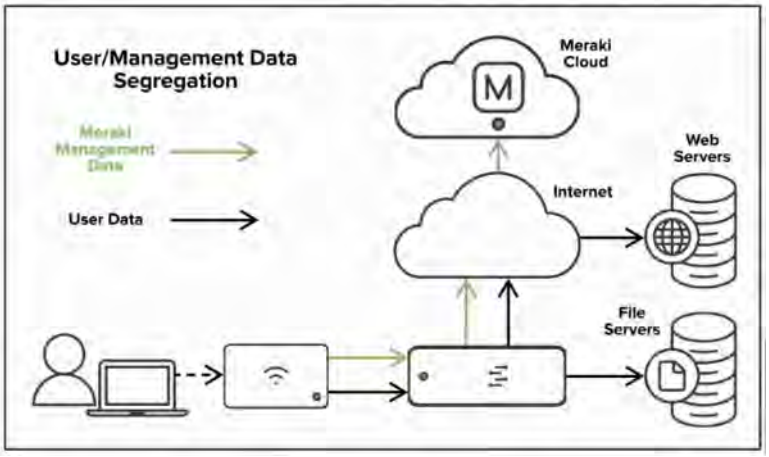
- SNIFFER
  - The AP does NOT OFFER a BSS for CLIENTS
  - Dedicated to CAPTURING 802.11 FRAMES and SENDING them to a DEVICE running software such as WIRESHARK
- MONITOR
  - The AP does NOT OFFER a BSS for CLIENTS
  - Dedicated to RECEIVING 802.11 FRAMES to detect ROGUE DEVICES
  - If a CLIENT is found to be a ROGUE DEVICE, an AP can send DE-AUTHENTICATION MESSAGES to disassociate the ROGUE DEVICE from the AP
- ROGUE DETECTOR
  - The AP does not even USE its RADIO
  - It LISTENS to traffic on the WIRED NETWORK only, but it receives a list of SUSPECTED ROGUE CLIENTS and AP MAC ADDRESSES from the WLC
  - By LISTENING to ARP MESSAGES on the WIRED NETWORK and correlating it with the information it receives from the WLC, it can DETECT ROGUE DEVICES
- SE-CONNECT (SPECTRUM EXPERT CONNECT)
  - The AP does NOT OFFER a BSS for CLIENTS
  - Dedicated to RF SPECTRUM ANALYSIS on ALL CHANNELS
  - It can send information to software such as Cisco Spectrum Expert on a PC to COLLECT and ANALYZE the DATA
- BRIDGE / MESH
  - Like the AUTONOMOUS APs *OUTDOOR BRIDGE* mode, the LIGHTWEIGHT AP can be a DEDICATED BRIDGE between SITES (Example: over LONG distances)
  - A MESH can be made between the ACCESS POINTS
- FLEX PLUS BRIDGE
  - Adds FLEXCONNECT functionality to the BRIDGE / MESH mode
  - Allows WIRELESS ACCESS POINTS to locally forward traffic even if connectivity to the WLC is lost



CLOUD-BASED APs

- CLOUD-BASED AP architecture is between AUTONOMOUS AP and SPLIT-MAC ARCHITECTURE
    - AUTONOMOUS APs that are centrally managed in the CLOUD
- CISCO MERAKI is a popular CLOUD-BASED WI-FI solution
- The MERAKI dashboard can be used to configure APs, monitor the NETWORK, generate performance reports, etc.
    - MERAKI also tells each AP which CHANNEL to use, what transmit power, etc.
- However, DATA TRAFFIC is not sent to the CLOUD. It is sent directly to the WIRED NETWORK like when using AUTONOMOUS APs
    - Only management / control traffic is sent to the CLOUD



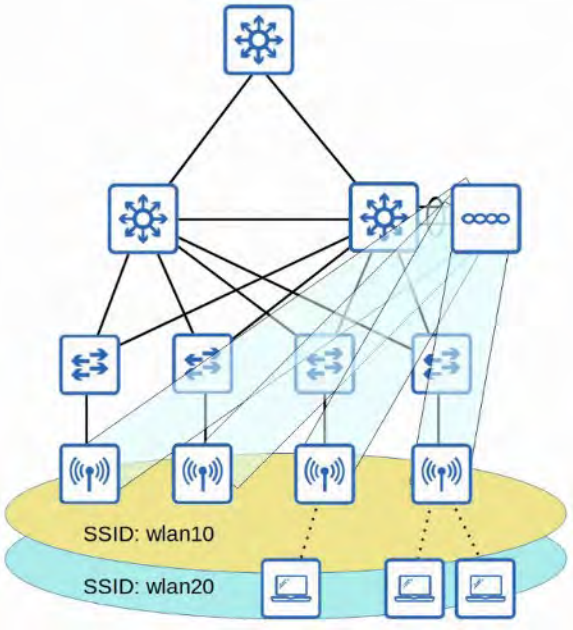


WIRELESS LAN CONTROLLER (WLC) DEPLOYMENTS

- In a SPLIT-MAC ARCHITECTURE, there FOUR MAIN WLC DEPLOYMENT MODES:
  - UNIFIED
    - THE WLC is a HARDWARE APPLICANCE in a central location of the NETWORK
  - CLOUD-BASED
    - The WLC is a VM running on a SERVER, usually in a PRIVATE CLOUD in a DATA CENTER
    - This is NOT the same as the CLOUD-BASED AP ARCHITECTURE discussed previously
  - EMBEDDED
    - The WLC is integrated within a SWITCH
  - MOBILITY EXPRESS
    - THE WLC is integrated within an AP

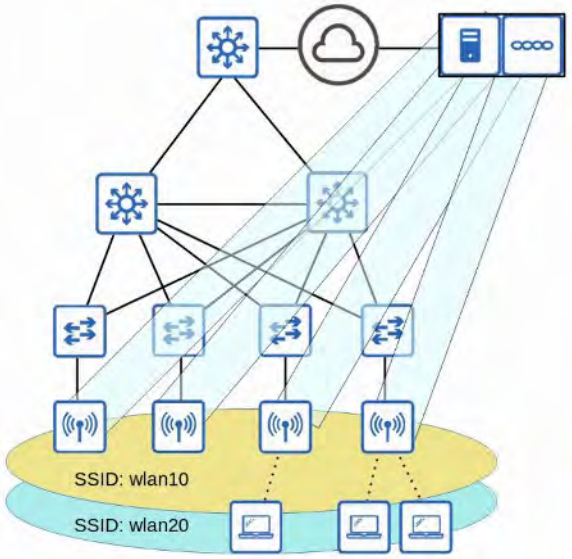
UNIFIED WLC

- THE WLC is a HARDWARE APPLICANCE in a central location of the NETWORK
- A UNIFIED WLC can support up to about 6000 APs
- If more than 6000 APs are needed, additional WLCs can be added to the NETWORK



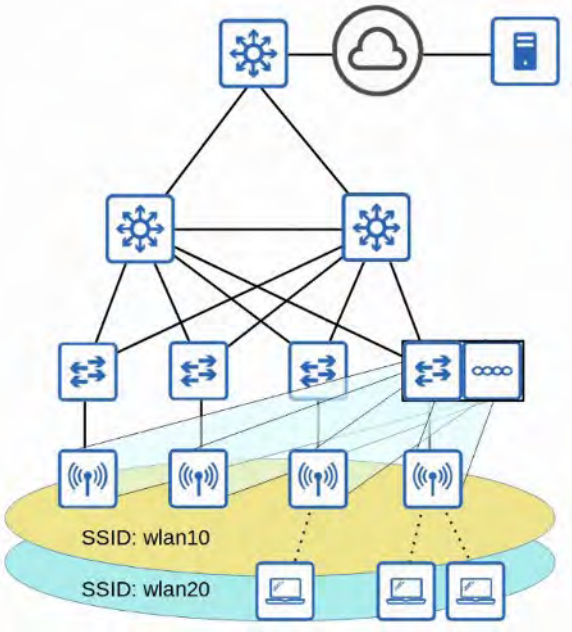
CLOUD-BASED

- The WLC is a VM running on a SERVER, usually in a PRIVATE CLOUD in a DATA CENTER
- CLOUD-BASED WLCs can typically support up to about 3000 APs
- If more than 3000 APs are needed, more WLC VMs can be deployed



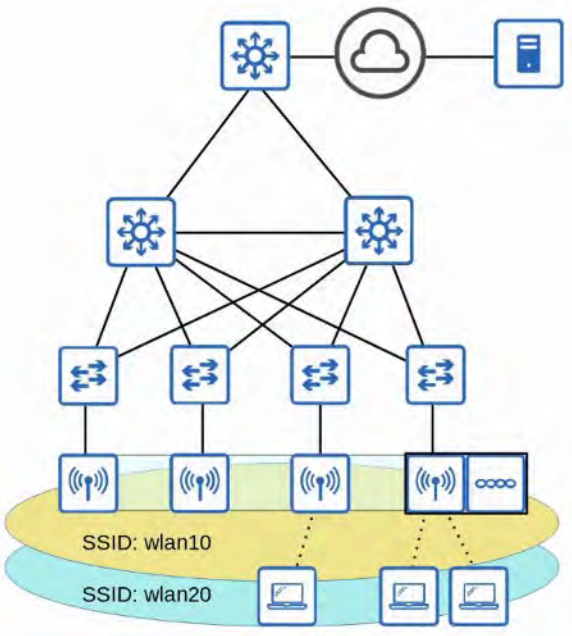
EMBEDDED WLC

- The WLC is embedded within a SWITCH
- An EMBEDDED WLC can support up to about 200 APs
- If more than 200 APs are needed, more SWITCHES with EMBEDDED WLCs can be added



CISCO MOBILITY EXPRESS WLC

- The WLC is embedded within an AP
- A MOBILITY EXPRESS WLC can support up to about 100 APs
- If more than 100 APs are needed, more APs with EMBEDDED MOBILITY EXPRESS WLCs can be added


SSID: wlan10
SSID: wlan20

57. WIRELESS SECURITY

INTRO TO WIRELESS NETWORK SECURITY

- Although SECURITY is important in ALL NETWORKS, it is even more essential in WIRELESS NETWORKS
- Because WIRELESS SIGNALS are not contained within a WIRE, any DEVICE within range of the signal can receive traffic
- In WIRED NETWORKS, traffic is often only ENCRYPTED when sent over an UNTRUSTED NETWORK such as the INTERNET
- In WIRELESS NETWORKS, it is VERY important to ENCRYPT traffic sent between the WIRELESS CLIENTS and the AP
- We will cover THREE MAIN CONCEPTS:
    - AUTHENTICATION
    - ENCRYPTION
    - INTEGRITY

AUTHENTICATION

- All CLIENTS must be AUTHENTICATED before they can associate with an AP
- In a corporate setting, only TRUSTED USERS / DEVICES should be given ACCESS to the NETWORK
    - In corporate settings, a separate SSID which doesn't have ACCESS to the corporate NETWORK can be provided for GUEST USERS
- Ideally, CLIENTS should also AUTHENTICATE the AP to avoid associating with a malicious AP
- There are MULTIPLE WAYS to AUTHENTICATE:
    - PASSWORD
    - USERNAME / PASSWORD
    - CERTIFICATES

ENCRYPTION

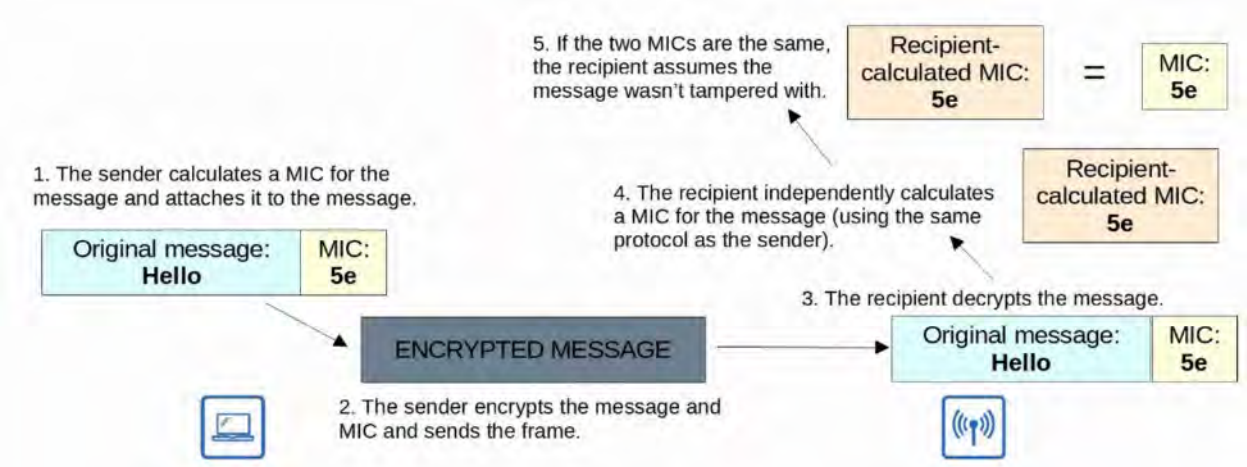
- Traffic sent between CLIENTS and APs should be ENCRYPTED so that it can't be read by anyone except the AP and the CLIENT
- There are many possible PROTOCOLS that can be used to ENCRYPT traffic
- All DEVICES on the WLAN will use the same PROTOCOL, however each CLIENT will use a unique ENCRYPTION / DECRYPTION KEY so that other DEVICES can't read its traffic
- A "GROUP KEY" is used by the AP to ENCRYPT traffic that it wants to send to all of its clients
    - All of the CLIENTS associated with the AP keep that key so they can DECRYPT the traffic

---

INTEGRITY

- As explained in the "SECURITY FUNDAMENTALS" video of the course, INTEGRITY ensures that the message is not modified by a third-party

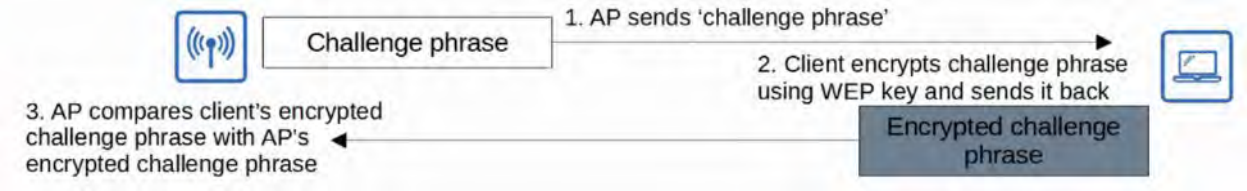
- The message that is sent by the SOURCE HOST should be the same as the message that is received by the DESTINATION HOST
- A MIC (Message Integrity Check) is added to the message to help protect their INTEGRITY.



AUTHENTICATION METHODS

The original 802.11 STANDARD included TWO OPTIONS for AUTHENTICATION:

- OPEN AUTHENTICATION
  - The CLIENT sends an AUTHENTICATION REQUEST and the AP just accepts it
  - The is clearly NOT a SECURE AUTHENTICATION method
  - After the CLIENT is AUTHENTICATED and associated with the AP, it's possible to require the USER to AUTHENTICATE via other methods before ACCESS to the NETWORK is granted (ie: Starbucks WI-FI)
- WEP (Wired Equivalent Privacy)
  - WEP is used to provide both AUTHENTICATION and ENCRYPTION of WIRELESS traffic
  - For ENCRYPTION, WEP uses the RC4 ALGORITHM
  - WEP is a "SHARED-KEY" PROTOCOL, requiring the SENDER and RECEIVER to have the same KEY
  - WEP KEYS can be 40 bits or 104 bits in length
  - The above KEYS are combined with a 24-bit "IV" (INITIALIZATION VECTOR) to bring the total length to 64 bits or 128 bits
  - WEP ENCRYPTION is NOT SECURE and can easily be cracked
  - WEP can be used for AUTHENTICATION like this:

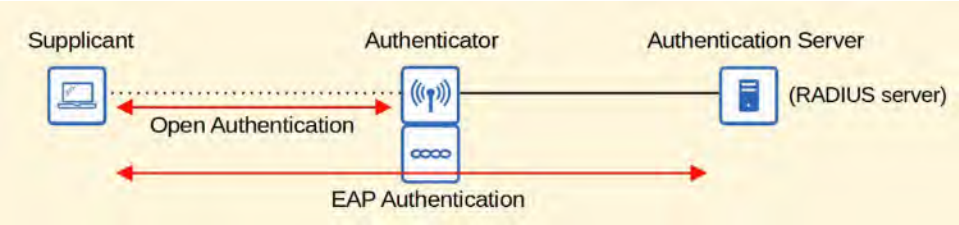


EAP (Extensible Authentication Protocol)

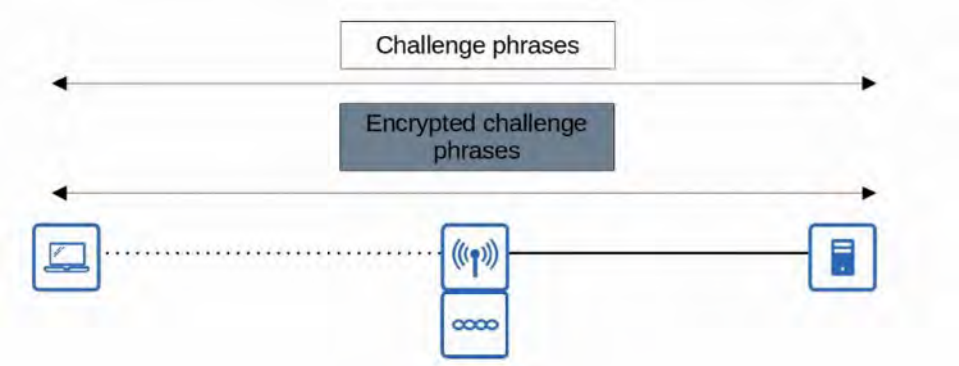
- EAP is an AUTHENTICATION FRAMEWORK
- It defines a STANDARD SET of AUTHENTICATION FUNCTIONS that are used by the various *EAP METHODS*
- We will look at FOUR EAP METHODS:
  - LEAP
  - EAP-FAST
  - PEAP
  - EAP-TLS
- EAP is integrated with **802.1X** which provides *PORT-BASED NETWORK ACCESS CONTROL*

802.1X is used to limit NETWORK ACCESS for CLIENTS connected to a LAN or WLAN until they AUTHENTICATE
There are **THREE MAIN ENTITIES** in 802.1X:

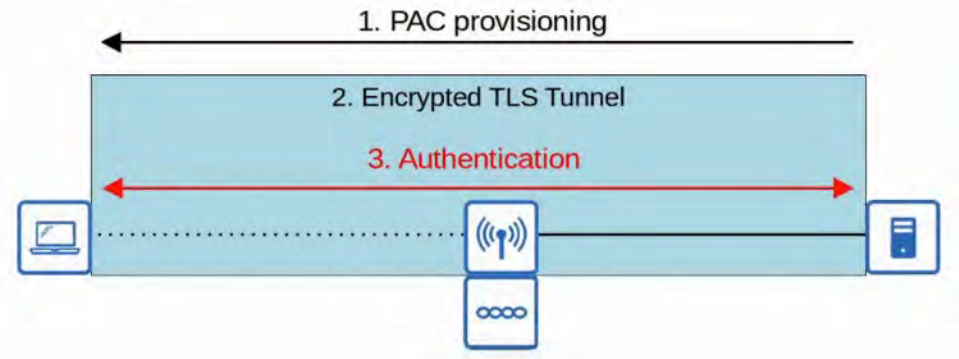
- SUPPLICANT : The DEVICE that wants to connect to the NETWORK
- AUTHENTICATOR : The DEVICE that provides access to the NETWORK
- AUTHENTICATION SERVER (AS) : The DEVICE that receives CLIENT credentials and PERMITS / DENIES ACCESS



- LEAP (Lightweight EAP)
  - LEAP was developed by Cisco an an improvement over WEP
  - CLIENTS must provide a USERNAME and PASSWORD to AUTHENTICATE
  - In addition, *MUTUAL AUTHENTICATION* is provided by both the CLIENT and SERVER sending a CHALLENGE PHRASE to each other.
  - DYNAMIC WEP KEYS are used, meaning that the WEP KEYS are changed frequently
  - Like WEP, LEAP is considered vulnerable and should not be used anymore

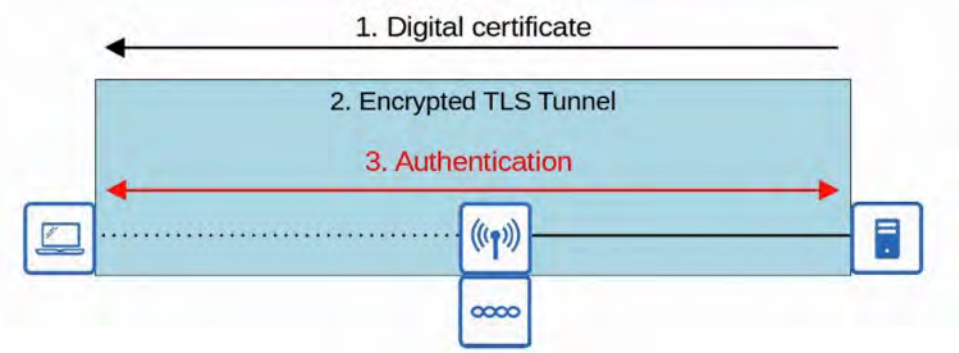


- EAP-FAST (EAP FLEXIBLE AUTHENTICATION via SECURE TUNNELING)
  - EAP-FAST was also developed by Cisco
  - Consists of THREE PHASES:
    - A PAC (PROTECTED ACCESS CREDENTIAL) is generated and passed from SERVER to CLIENT
    - A SECURE TLS TUNNEL is established between the CLIENT and AUTHENTICATION SERVER
    - Inside of the SECURE (ENCRYPTED) TLS TUNNEL, the CLIENT and SERVER communicated further to AUTHENTICATE / AUTHORIZE the CLIENT

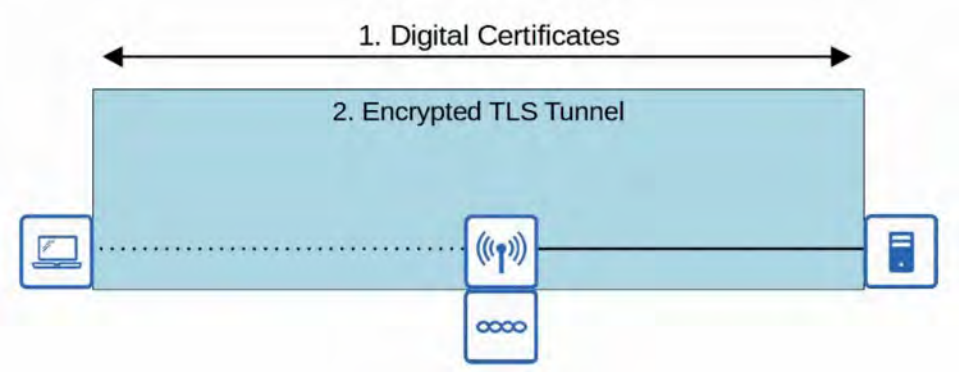


- PEAP (PROTECTED EAP)

- Like EAP-FAST, PEAP involves establishing a SECURE TLS TUNNEL between the CLIENT and SERVER
- Instead of a PAC, the SERVER has a DIGITAL CERTIFICATE
- The CLIENT uses this DIGITAL CERTIFICATE to AUTHENTICATE the SERVER
- The CERTIFICATE is also used to establish a TLS TUNNEL
- Because only the SERVER provides a CERTIFICATE for AUTHENTICATION, the CLIENT must still be AUTHENTICATED within the SECURE TUNNEL
  - Example: MS-CHAP (Microsoft Challenge-Handshake Authentication Protocol)



- EAP-TLS (EAP TRANSPORT LAYER SECURITY)
  - Whereas PEAP only requires the AS to have a CERTIFICATE, EAP-TLS requires a CERTIFICATE on the AS and on every single CLIENT
  - EAP-TLS is the MOST SECURE WIRELESS AUTHENTICATION method, but it is more difficult to implement than PEAP because every CLIENT DEVICE needs a CERTIFICATE
  - Because the CLIENT and SERVER AUTHENTICATE each other with DIGITAL CERTIFICATES, there is no need to AUTHENTICATE the CLIENT within the TLS TUNNEL
  - The TLS TUNNEL is still used to exchange ENCRYPTION KEY information (ENCRYPTION methods will be discussed next)



ENCRYPTION / INTEGRITY METHODS

- TKIP (Temporal Key Integrity Protocol)
  - WEP was found to be vulnerable, but WIRELESS hardware at the time was built to use WEP
  - A temporary solution was needed until a new STANDARD was created and a new HARDWARE was built
  - TKIP adds various SECURITY FEATURES:
    - A MIC (Message Integrity Check) is added to protect the integrity of messages
    - A KEY MIXING ALGORITHM is used to create a unique WEP key for every frame
    - The INITIALIZATION VECTOR is doubled in length from 24 bits to 48 bits, making BRUTE-FORCE attacks much more difficult
    - The MIC includes the SENDER MAC ADDRESS to identify the FRAME's SENDER

        - A TIMESTAMP is added to the MIC to prevent replay attacks. Replay attacks involved re-resending a FRAME that has already been transmitted
        - A TKIP SEQUENCE NUMBER is used to keep track of FRAMES sent from each SOURCE MAC ADDRESS. This also protects against REPLAY ATTACKS

** You probably don't need to memorize ALL of the above features
** TKIP is used in WPA version 1, which will be discussed in the next section

- CCMP (Counter / CBC-MAC Protocol)
    - CCMP was developed after TKIP and is more SECURE
    - It is used in WPA2
    - To use CCMP, it must be supported by the DEVICE'S hardware.
    - Old hardware built only to use WEP / TKIP cannot use CCMP
    - CCMP consists of TWO DIFFERENT ALGORITHMS to provide ENCRYPTION and MIC :
        - AES (Advanced Encryption Standard) COUNTER MODE ENCRYPTION
            - AES is the MOST SECURE ENCRYPTION PROTOCOL currently available.
            - Widely used all over the world
            - There are multiple MODES of operation for AES.
            - CCMP uses "COUNTER MODE"
        - CBC-MAC (CIPHER BLOCK CHAINING MESSAGE AUTHENTICATION CODE)
            - Used as a MIC to ENSURE the INTEGRITY of MESSAGES
- GCMP (GALOIS / COUNTER MODE PROTOCOL)
    - GCMP is MORE SECURE and EFFICIENT than CCMP
    - Its increased efficiency allows higher data throughput than CCMP
    - It is used in WPA3
    - GCMP consists of TWO ALGORITHMS:
        - AES COUNTER MODE ENCRYPTION
        - GMAC (GALOIS MESSAGE AUTHENTICATION CODE)
            - Used as a MIC to ENSURE the INTEGRITY of MESSAGE

WI-FI PROTECTED ACCESS (WPA)

- The WI-FI Alliance has developed THREE WPA CERTIFICATIONS for WIRELESS DEVICES:
    - WPA
    - WPA2
    - WPA3
- To be WPA-CERTIFIED, EQUIPMENT must be TESTED in authorized testing labs
- All of the above support TWO AUTHENTICATION MODES:
    - PERSONAL MODE :
        - A PRE-SHARED KEY (PSK) is used for AUTHENTICATOIN
        - When you connect to a home WI-FI NETWORK, enter the PASSWORD and are AUTHENTICATED, that is PERSONAL MODE
        - This is common in small NETWORKS
        - The PSK itself is NOT sent over the air
        - A FOUR-WAY HANDSHAKE is used for AUTHENTICATION and the PSK is used to GENERATE ENCRYPTION KEYS
    - ENTERPRISE MODE :
        - 802.1X is used with an AUTHENTICATION SERVER (RADIUS SERVER)
        - No specific EAP METHOD is specified, so all are supported (PEAP, EAP-TLS, etc)

WPA

- The WPA CERTIFICATION was developed after WEP was proven to be vulnerable and includes the following PROTOCOLS:
    - TKIP (based on WEP) provides ENCRYPTION / MIC
    - 802.1X AUTHENTICATION (ENTERPRISE MODE) or PSK (PERSONAL MODE)

WPA2

- Was released in 2004 and includes the following PROTOCOLS:
    - CCMP provides ENCRYPTION / MIC

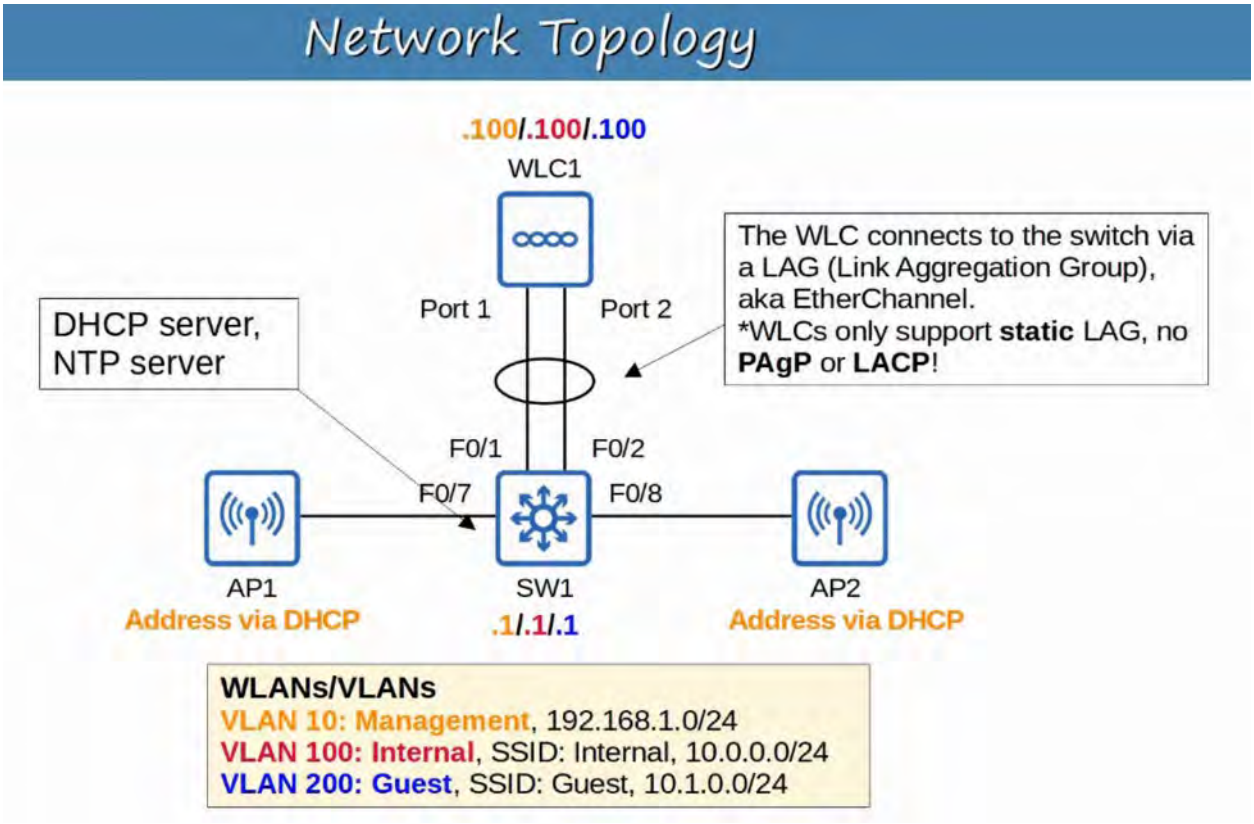
        - 802.1X AUTHENTICATION (ENTERPRISE MODE) or PSK (PERSONAL MODE)

WPA3

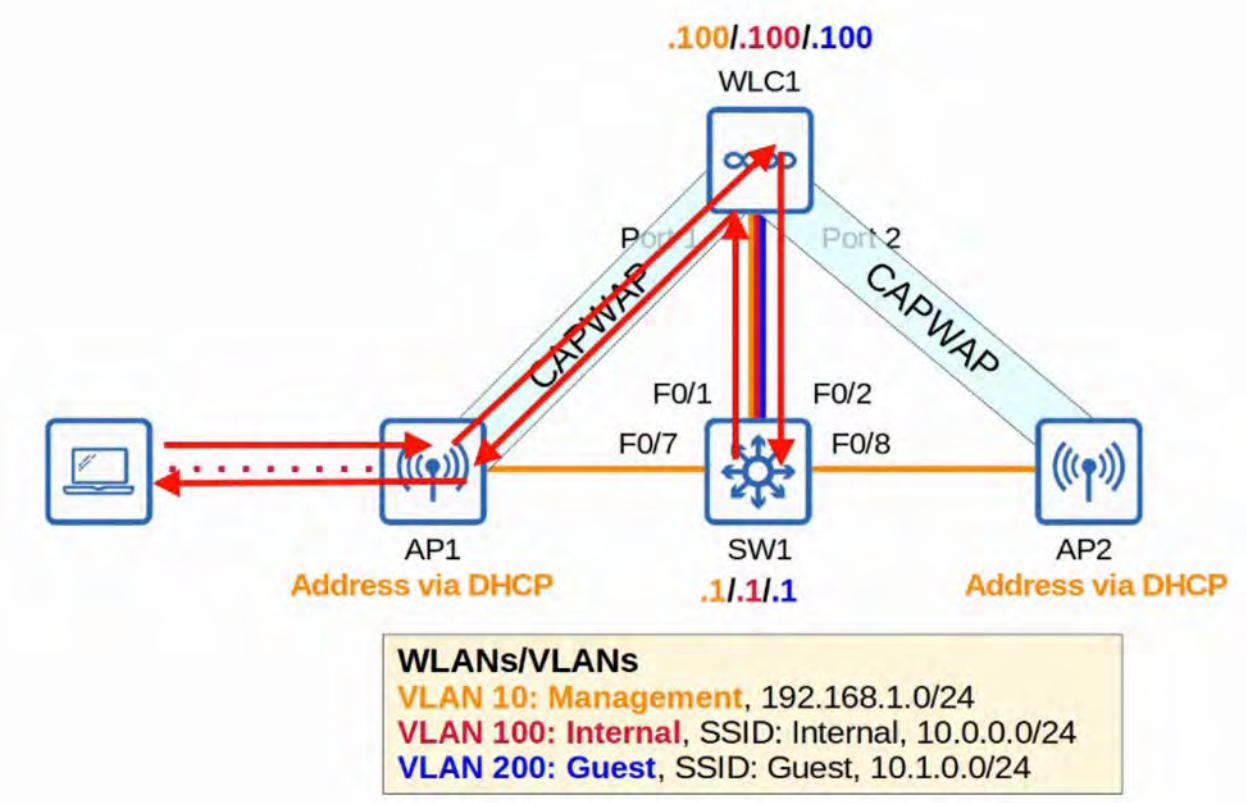
- Was released in 2018 and includes the following PROTOCOLS:
    - GCMP provides ENCRYPTION / MIC
    - 802.1X AUTHENTICATION (ENTERPRISE MODE) or PSK (PERSONAL MODE)
    - WPA3 also provides several additional security features:
        - PMF (PROTECTED MANAGEMENT FRAMES)
            - Protecting 802.11 MANAGEMENT FRAMES from eavesdropping / forging
        - SAE (SIMULTANEOUS AUTHENTICATION OF EQUALS)
            - Protects the four-way handshake when using PERSONAL MODE AUTHENTICATION
        - FORWARD SECRECY
            - Prevents DATA from being DECRYPTED after it has been transmitted over the air so an ATTACKER can’t capture WIRELESS FRAMES and then try to DECRYPT them later

58. WIRELESS CONFIGURATION

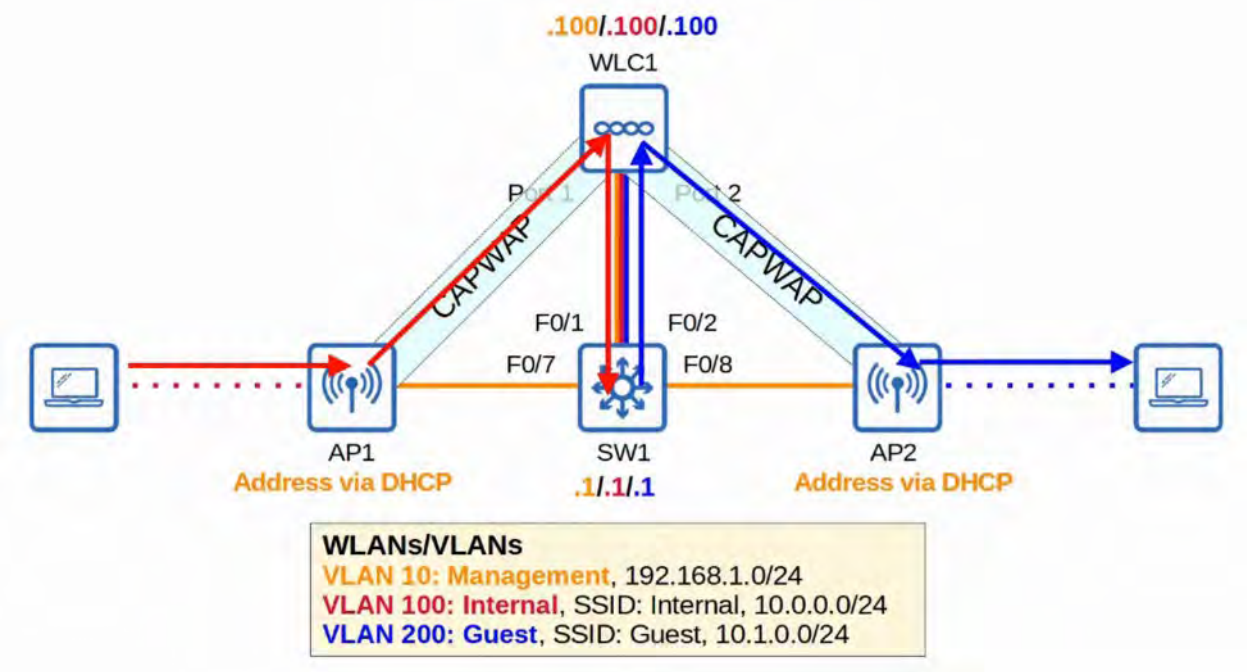
TOPOLOGY INTRODUCTION



INTERNAL PC (VLAN 100) ACCESSING DEFAULT GATEWAY via Internal CAPWAP tunnel



REACHING External GUEST PC via DEFAULT GATEWAY + Internal and External CAPWAP tunnels



LAYER 3 SWITCH CONFIGURATION (SW1)

```
SW1(config)#vlan 10
SW1(config-vlan)#name Management
SW1(config-vlan)#vlan 100
SW1(config-vlan)#name Internal
SW1(config-vlan)#vlan 200
SW1(config-vlan)#name Guest

SW1(config)#int range f0/6 - 8
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
SW1(config-if-range)#spanning-tree portfast

SW1(config-if-range)#interface range f0/1 - 2
SW1(config-if-range)#channel-group 1 mode on

SW1(config-if-range)#interface port-channel 1
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 10,100,200
```

I included F0/6 because I will connect my PC to F0/6 to gain access to WLC1's GUI.

Remember that WLCs only support static LAG, no PAgP or LACP.

PART 2 of configuration
Note DHCP "Option 43"

```
SW1(config)#interface vlan 10
SW1(config-if)#ip address 192.168.1.1 255.255.255.0
SW1(config-if)#interface vlan 100
SW1(config-if)#ip address 10.0.0.1 255.255.255.0
SW1(config-if)#interface vlan 200
SW1(config-if)#ip address 10.1.0.1 255.255.255.0

SW1(config)#ip dhcp pool VLAN10
SW1(dhcp-config)#network 192.168.1.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.1.1
SW1(dhcp-config)#option 43 ip 192.168.1.100

SW1(config)#ip dhcp pool VLAN100
SW1(dhcp-config)#network 10.0.0.0 255.255.255.0
SW1(dhcp-config)#default-router 10.0.0.1

SW1(config)#ip dhcp pool VLAN200
SW1(dhcp-config)#network 10.1.0.0 255.255.255.0
SW1(dhcp-config)#default-router 10.1.0.1

SW1(config)#ntp master
```

Option 43 can be used to tell the APs the IP address of their WLC.
*this is not necessary in this case because the APs and WLC are in the same subnet. The WLC will hear the APs broadcast CAPWAP *discovery* messages.

WLC SETUP
This helps set up the WLC to allow GUI configuration

WLC Initial Setup

```
Welcome to the Cisco Wizard Configuration Tool
Use the '-' character to backup


Would you like to terminate autoinstall? [yes]:

System Name [Cisco_10:65:64] (31 characters max): WLC1
Enter Administrative User Name (24 characters max): admin
Enter Administrative Password (3 to 24 characters): *********
Re-enter Administrative Password                  : *********

Enable Link Aggregation (LAG) [yes][NO]: yes

Management Interface IP Address: 192.168.1.100
Management Interface Netmask: 255.255.255.0
Management Interface Default Router: 192.168.1.1
Management Interface VLAN Identifier (0 = untagged): 10
Management Interface DHCP Server IP Address: 192.168.1.1
```

```
Virtual Gateway IP Address: 172.16.1.1

Multicast IP Address: 239.239.239.239

Mobility/RF Group Name: jITlab

Network Name (SSID): Internal

Configure DHCP Bridging Mode [yes][NO]: no

Allow Static IP Addresses [YES][no]: yes

Configure a RADIUS Server now? [YES][no]: no
Warning! The default WLAN security policy requires a RADIUS server.
Please see documentation for more details.

Enter Country Code list (enter 'help' for a list of countries) [US]: FR
```

We will change the WLAN security policy to PSK, so we don't need to configure a RADIUS server.

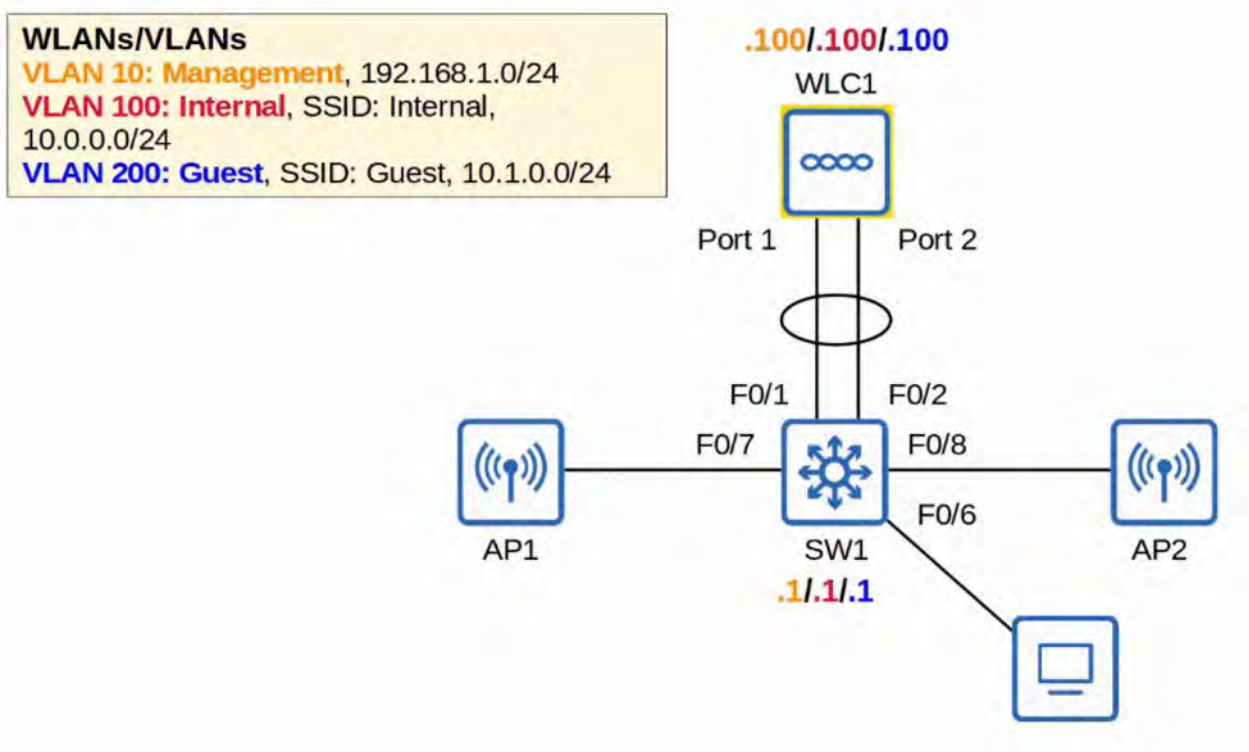
Why Jeremy chose FRANCE for Country Code (has to do with regulatory domain of equipment)

AIR-CAP3502I-E-K9

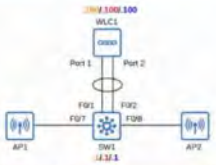
- -E is the *regulatory domain* of the device.
- -E indicates Europe.
- If the regulatory domain of the country specified in the WLC configuration doesn't match the regulatory domain of the AP, the AP won't be able to join the WLC.
- https://www.cisco.com/c/dam/assets/prod/wireless/wireless-compliance-tool/index.html to check the regulatory domain of each country.

```
Enable 802.11b Network [YES][no]:
Enable 802.11a Network [YES][no]:
Enable 802.11g Network [YES][no]:
Enable Auto-RF [YES][no]:

Configure a NTP server now? [YES][no]: yes
Enter the NTP server's IP address: 192.168.1.1
Enter a polling interval between 3600 and 604800 secs: 3600

Configuration correct? If yes, system will save it and reset. [yes][NO]:
yes

Configuration saved!
Resetting system with new configuration...
```

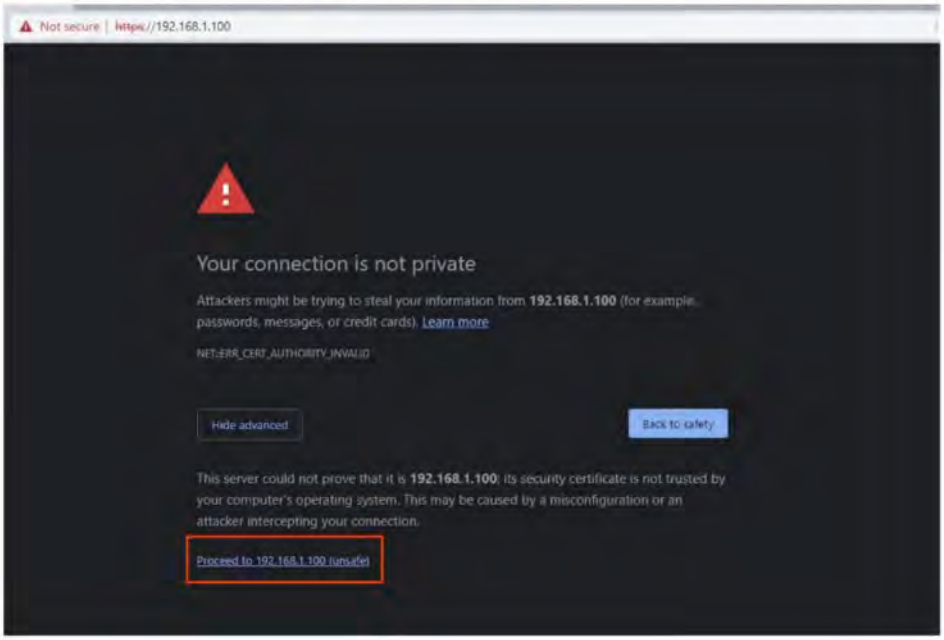
ACCESSING THE WLC GUI

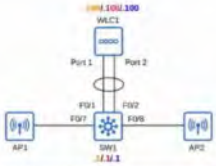


Accessing the GUI

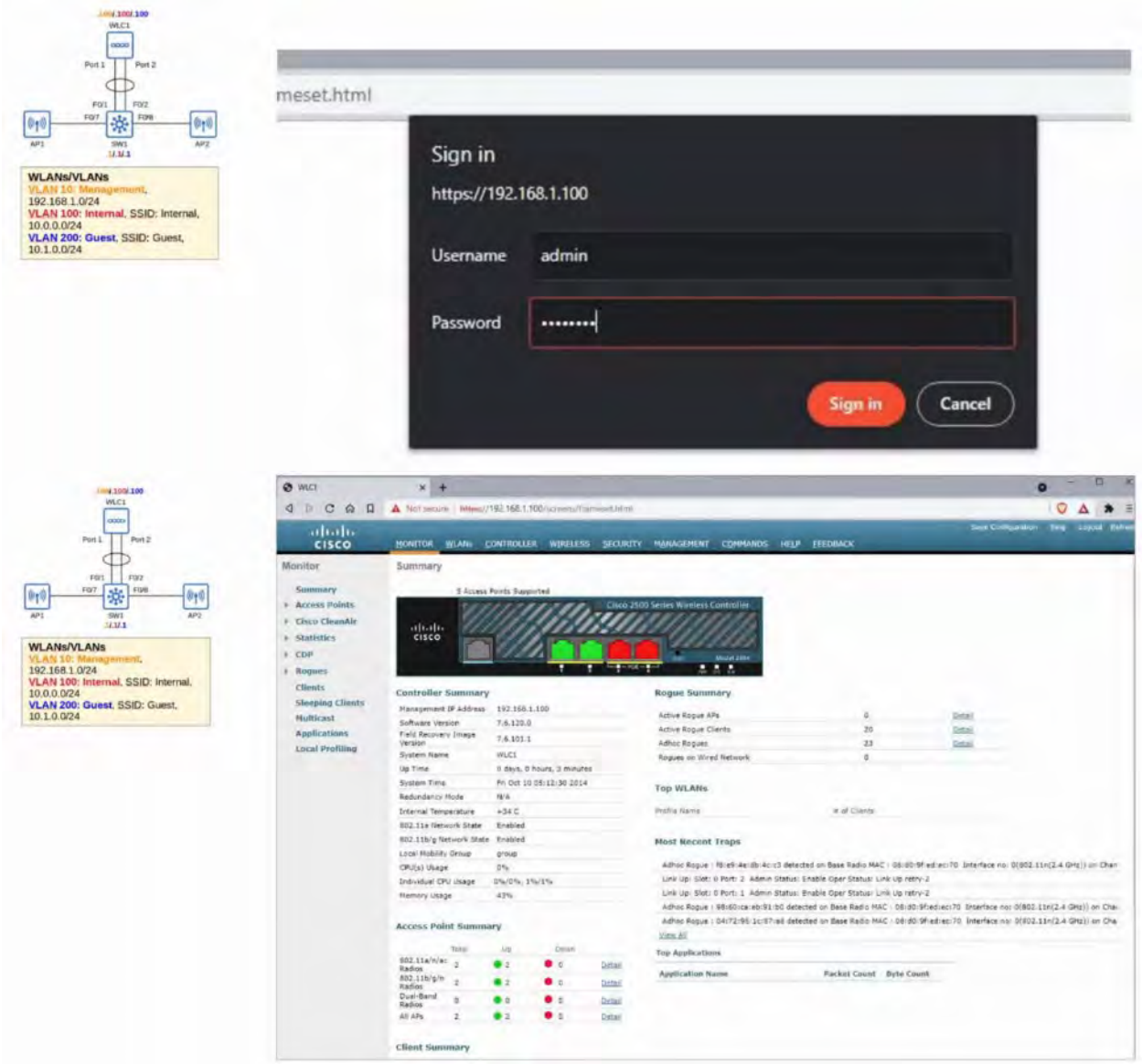


WLANs/VLANs
VLAN 10: Management, 192.168.1.0/24
VLAN 100: Internal, SSID: Internal, 10.0.0.0/24
VLAN 200: Guest, SSID: Guest, 10.1.0.0/24





WLANs/VLANs
VLAN 10: Management, 192.168.1.0/24
VLAN 100: Internal, SSID: Internal, 10.0.0.0/24
VLAN 200: Guest, SSID: Guest, 10.1.0.0/24

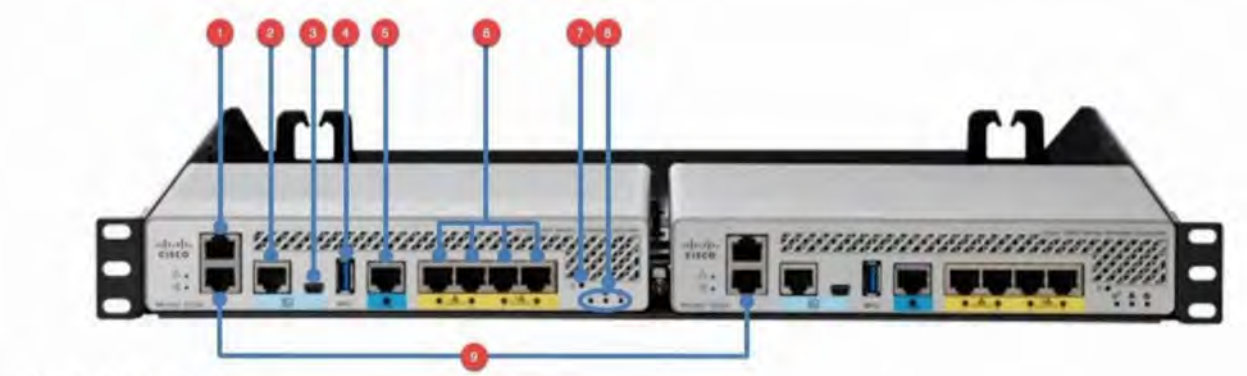


WLC CONFIGURATION

WLC PORTS

- WLC PORTS are the PHYSICAL PORTS that cables connect to

- WLC INTERFACES are the logical interfaces within the WLC (ie: SVIs on a SWITCH)
- WLCs have a few different PORTS:
    - SERVICE PORT
        - A dedicated MANAGEMENT PORT
        - Used for OUT-OF-BAND management
        - Must connected to a SWITCH ACCESS PORT because it only supports one VLAN
        - This PORT can be used to connect to the DEVICE while it is booting, performing system recovery, etc.
    - DISTRIBUTION SYSTEM PORT
        - These are the standard NETWORK PORTS that connect to the “DISTRIBUTION SYSTEM” (WIRED NETWORK) and are used for DATA traffic.
        - These PORTS usually connect to SWITCH TRUNK PORTS, and if multiple distribution PORTS are used they can form a LAG
    - CONSOLE PORT
        - This is a standard CONSOLE PORT, either RJ45 or USB
    - REDUNDANCY PORT
        - This PORT is used to connect to another WLC to form a HIGH AVAILABILITY (HA) pair



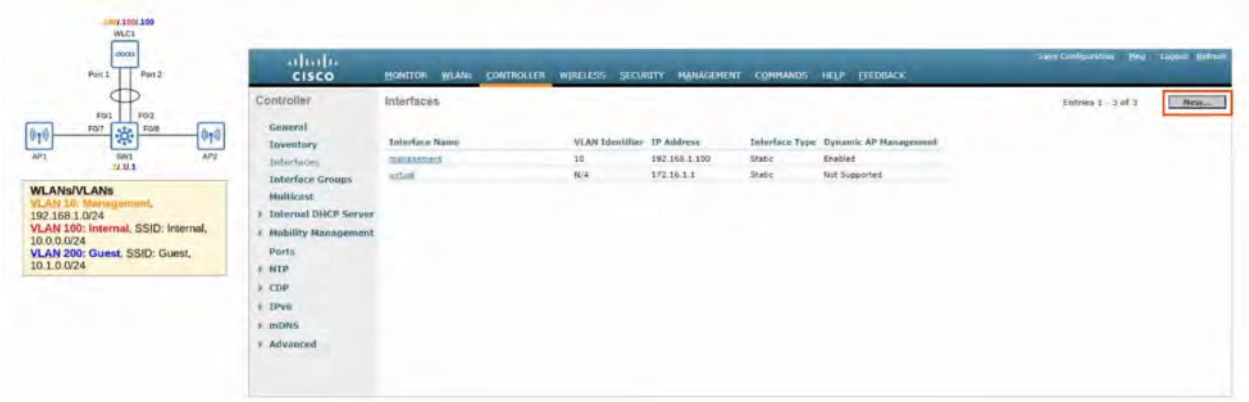
1) Service port
2) Console port (RJ45)
3) Console port (USB)
4) USB (for software updates)
5) Distribution system port (multi-gigabit)
6) Distribution system ports (1-gig)
7) Reset button
8) Status LEDs
9) Redundancy port

WLC INTERFACES

- MANAGEMENT INTERFACES
    - Used for management traffic such as TELNET, SSH, HTTP, HTTPS, RADIUS authentication, NTP, SYSLOG, etc.
    - CAPWAP TUNNELS are also formed to / from the WLC’s management INTERFACE
- REDUNDANCY MANAGEMENT INTERFACE
    - When TWO WLCs are connected by their REDUNDANCY PORTS, one WLC is “ACTIVE” and the other is “STANDBY”
    - This INTERFACE can be used to connect to and manage the “STANDBY” WLC
- VIRTUAL INTERFACE
    - This INTERFACE is used when communicating with WIRELESS CLIENTS to relay DHCP requests, perform CLIENT WEB AUTHENTICATION, etc.
- SERVICE PORT INTERFACE
    - If the SERVICE PORT is used, this INTERFACE is bound to it and used for OUT-OF-BAND MANAGEMENT

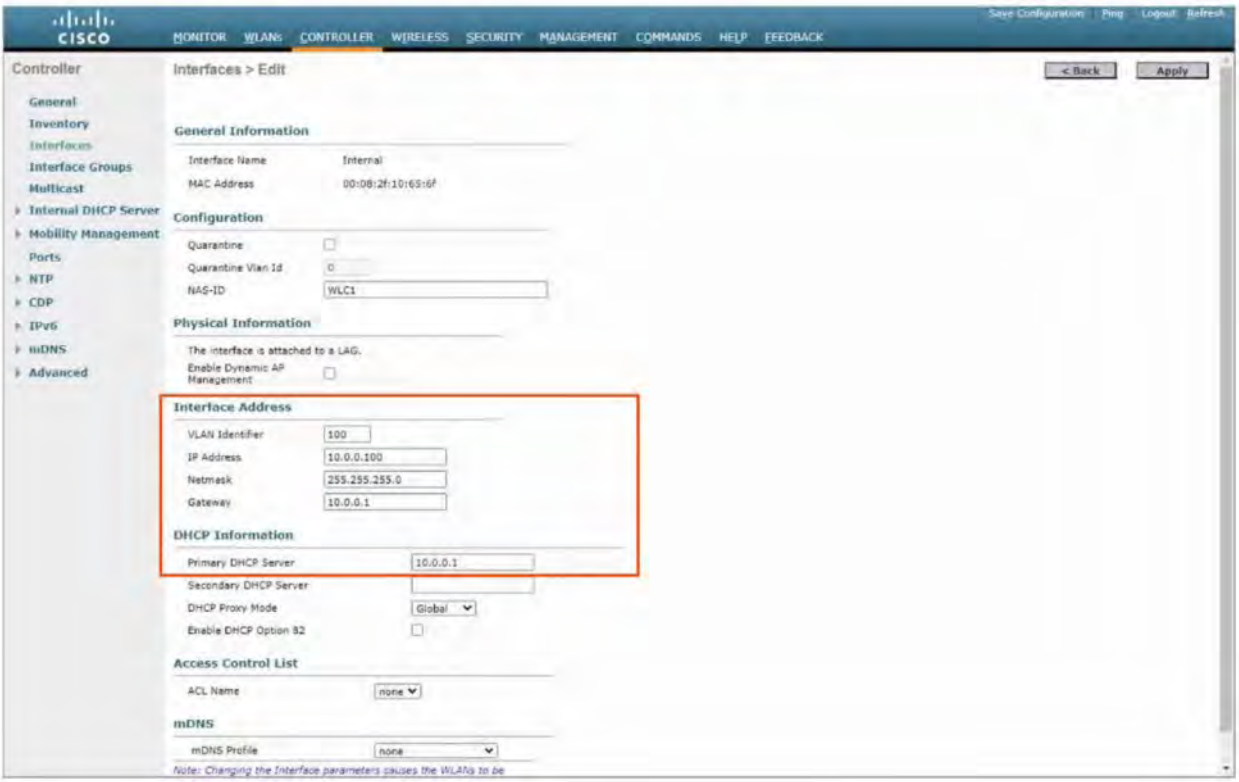
- DYNAMIC INTERFACE
    - These are the INTERFACES used to map a WLAN to a VLAN
    - For example :
        - TRAFFIC from the “INTERNAL” WLAN will be sent to the WIRED NETWORK from the WLCs “INTERNAL” DYNAMIC INTERFACE

WLAN CONFIGURATION
Click “NEW”

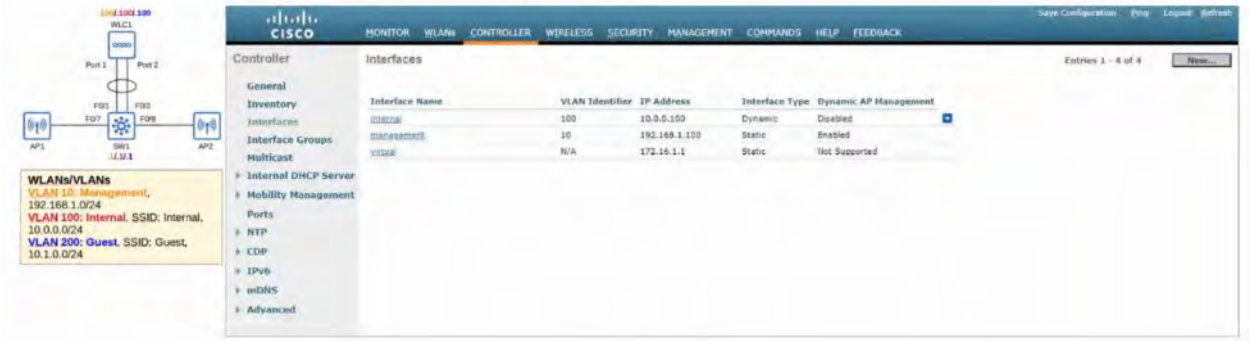


Fill in details of the interface and click “APPLY”

Fill out details (IP, Netmask, Gateway…) and then click “APPLY”

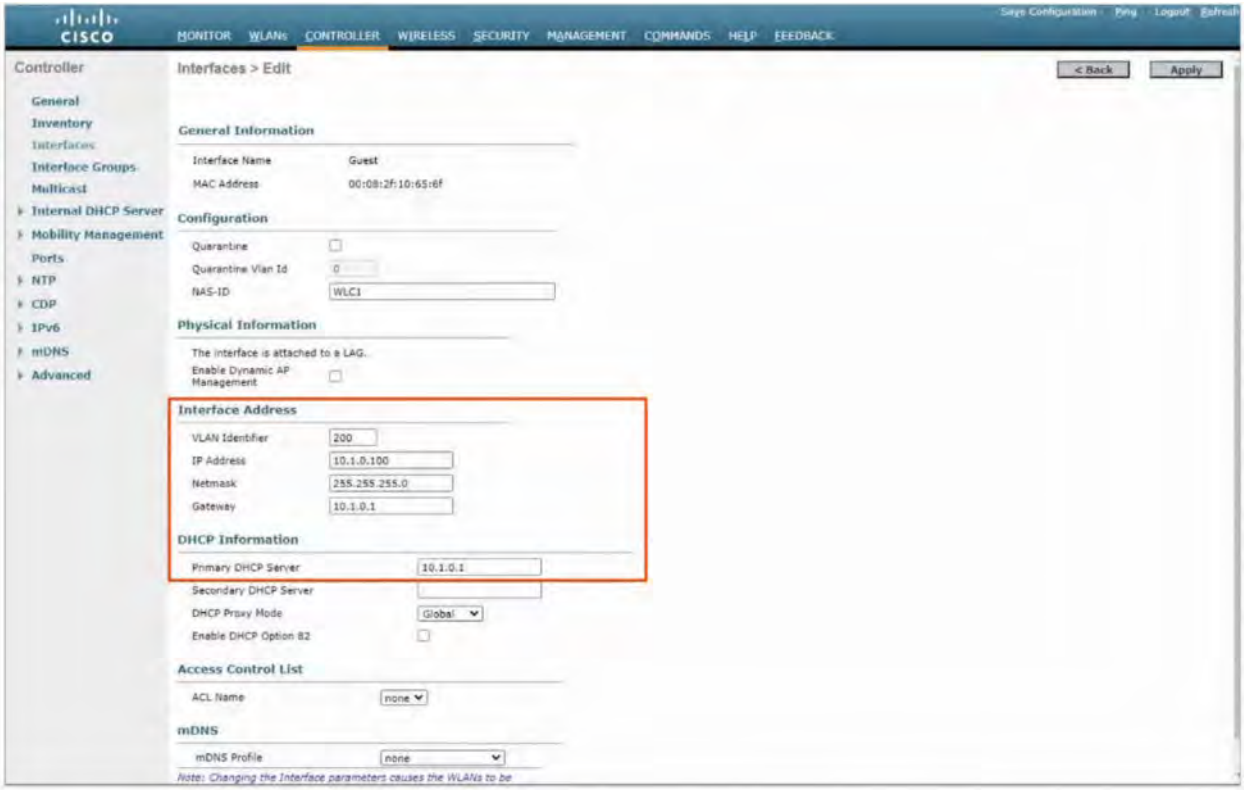


INTERNAL interface has now been created

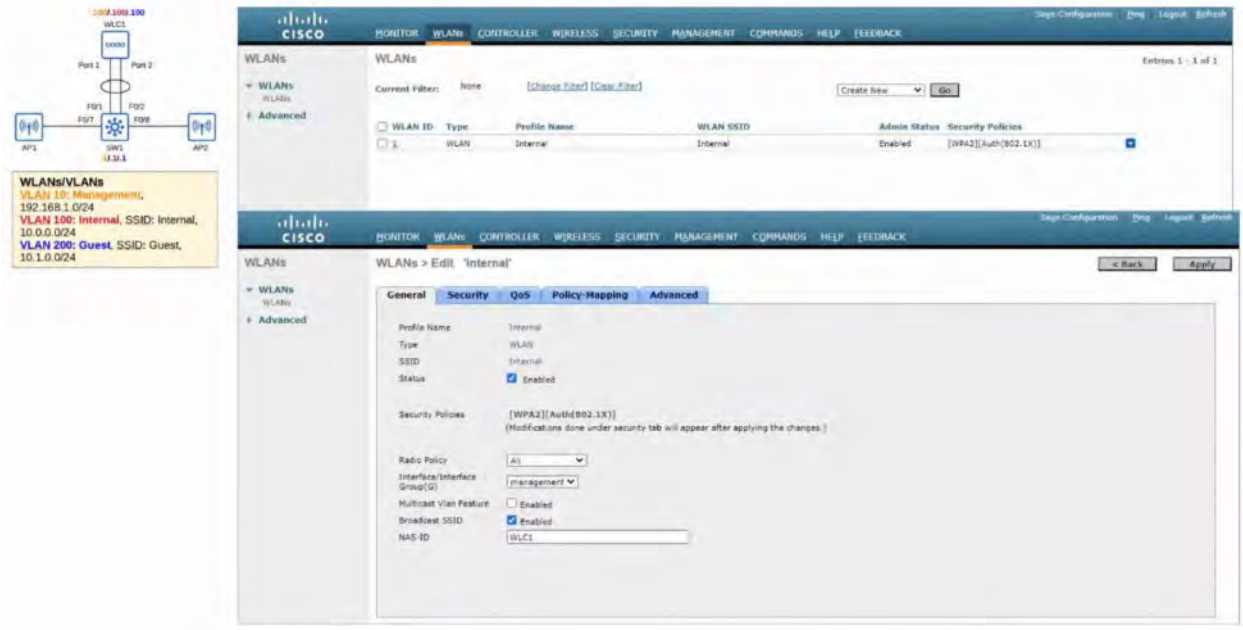


Now, repeat the above steps for the GUEST interface

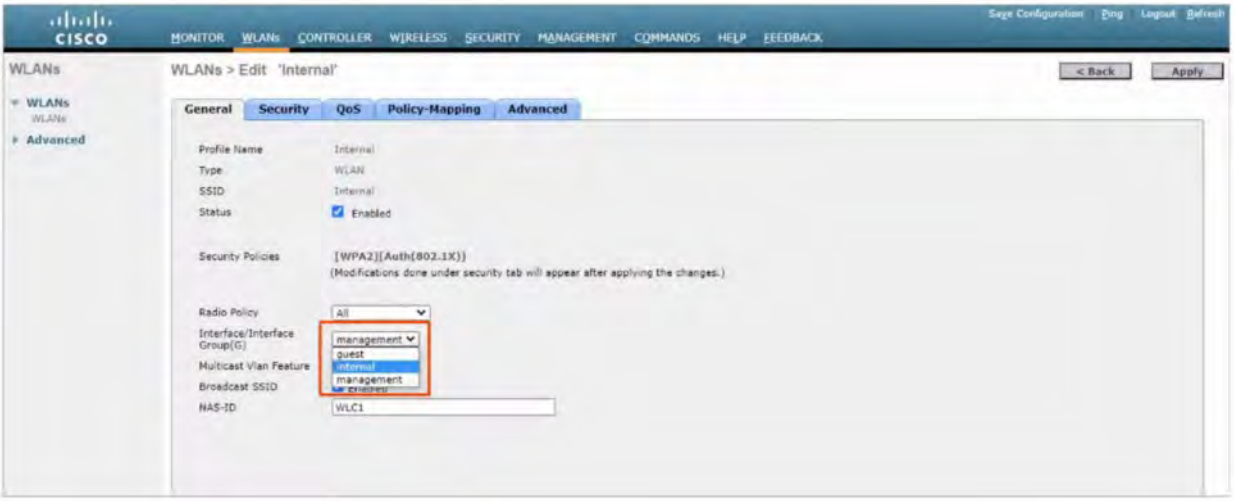
Fill out details (IP, Netmask, Gateway…) and then click “APPLY”



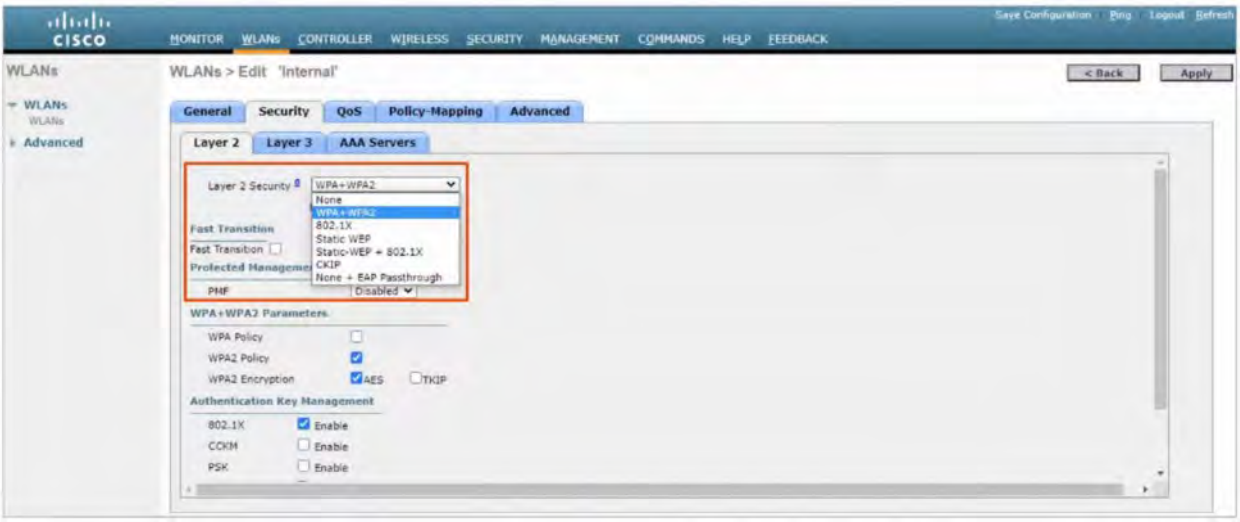
Now that all the INTERFACES are created, we can start WLAN CONFIGURATION



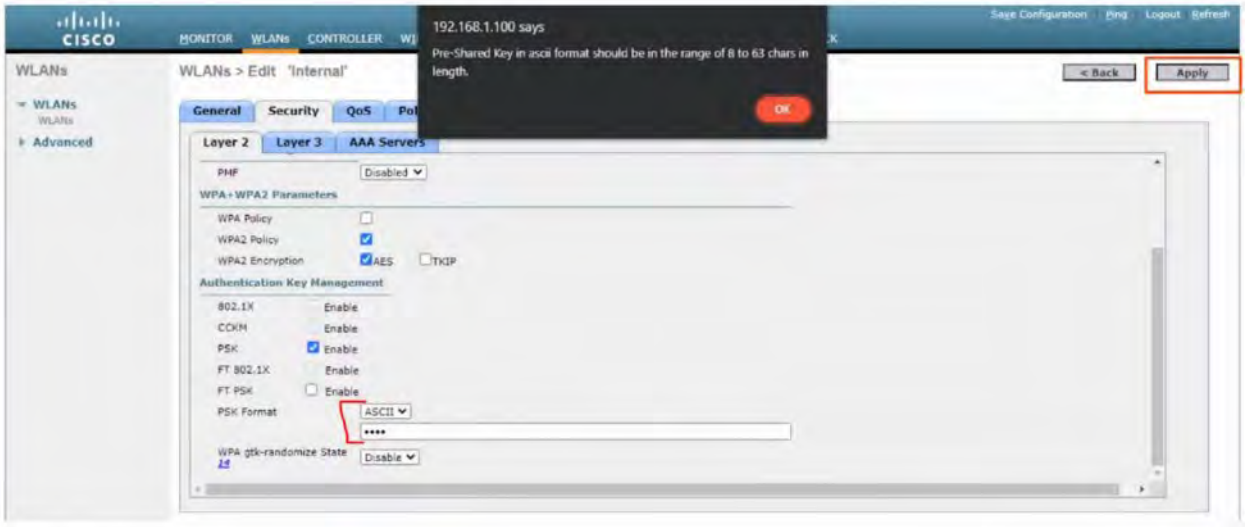
INTERNAL WLAN is set to “MANAGEMENT”, it needs to be changed to “INTERNAL”

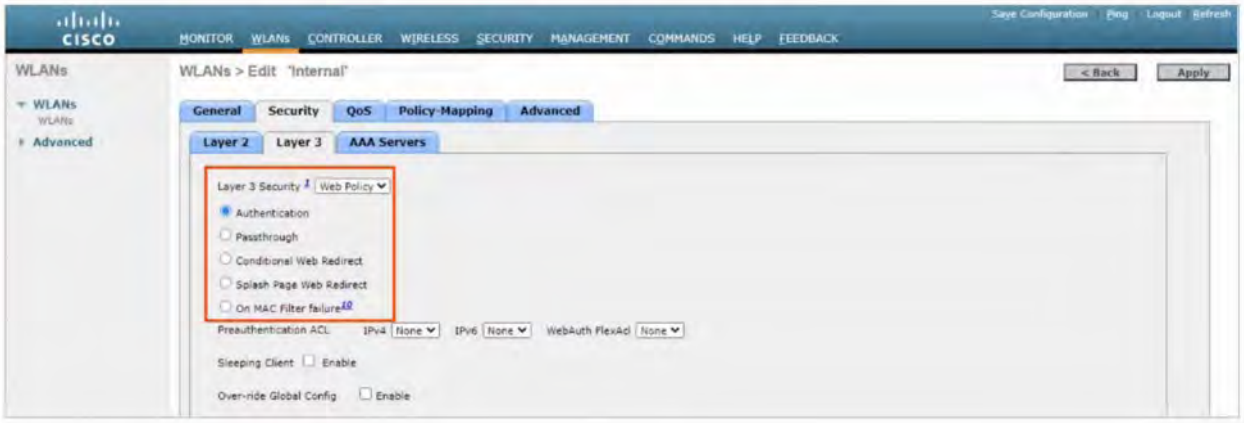


SECURITY will also need to be changed from [WPA2] to [WPA2 PSK]



(Need to CHECK the PSK “Enable” box at the bottom)
Change the PSK FORMAT to “ASCII” and enter a PASSWORD (at least 8 chars in length)

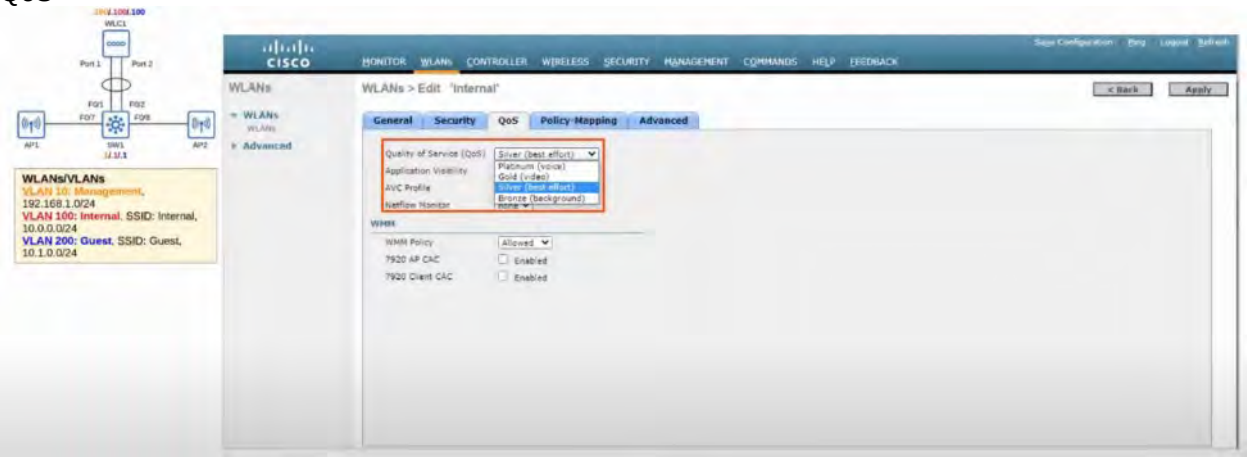




- WEB AUTHENTICATION
  - After the WIRELESS CLIENTS gets an IP ADDRESS and tries to access a WEB PAGE, they will have to enter a USERNAME and PASSWORD to AUTHENTICATE
- WEB PASSTHROUGH
  - Similar to the above, but NO USERNAME or PASSWORD are required

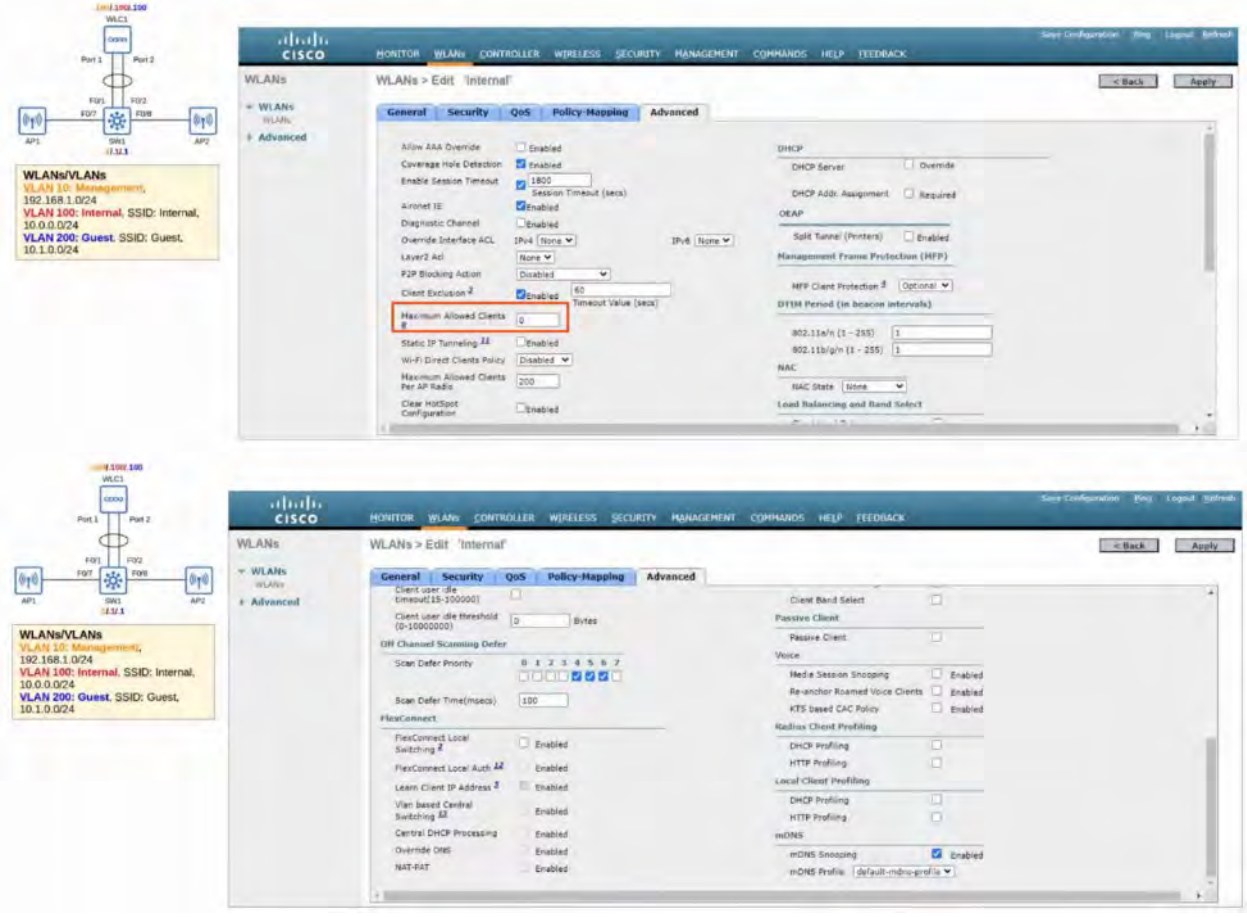
    - A warning or statement is displayed and the CLIENT simply has to agree to gain access to the INTERNET
- CONDITIONAL and SPLASH PAGE web redirect options are similar but additionally require 802.1x LAYER 2 AUTHENTICATION

QoS

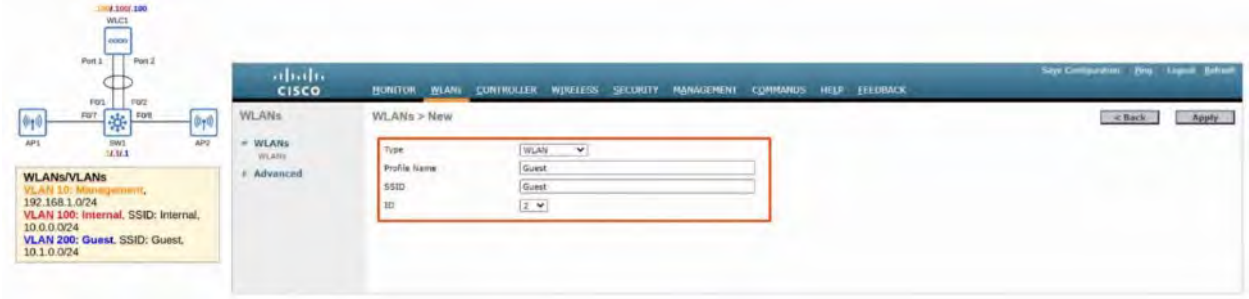


Default QoS setting is “SILVER” (Best Effort). This can be changed depending on the class of traffic being sent through the WLAN

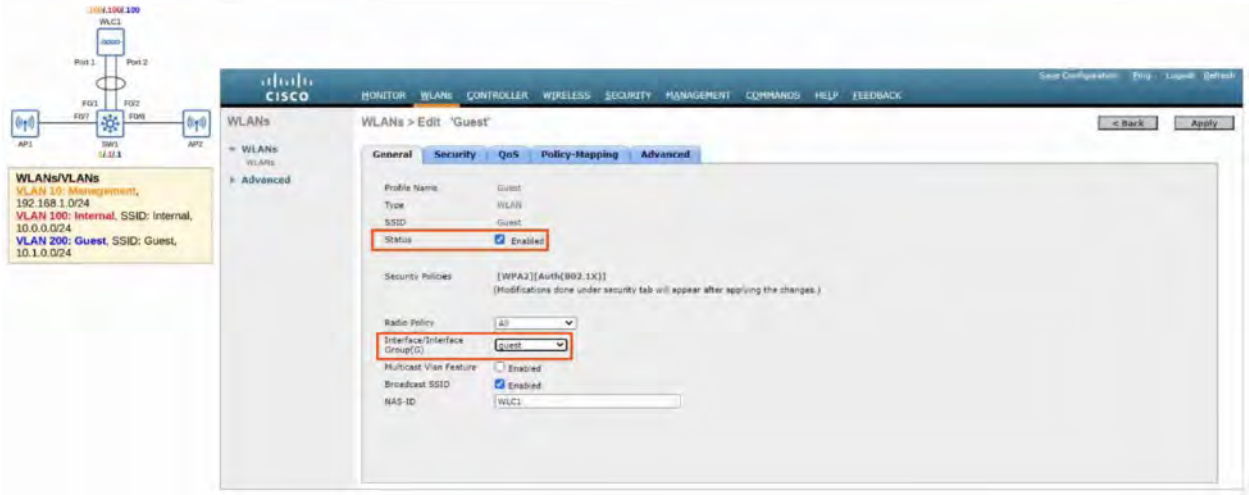
ADVANCED SETTINGS



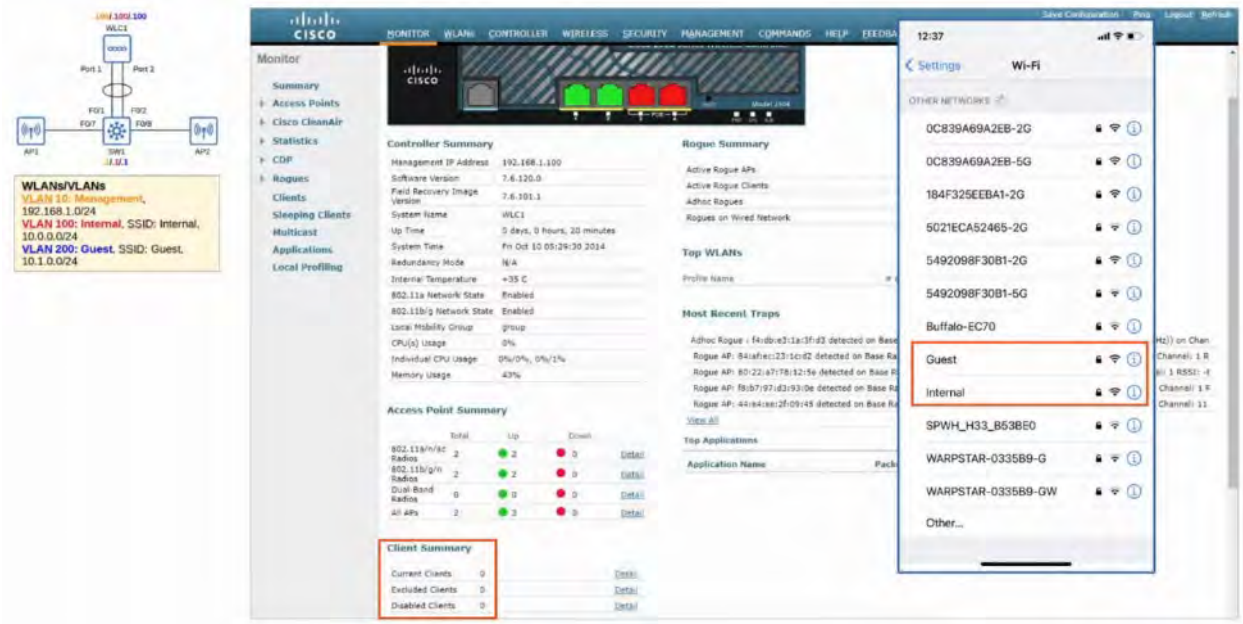
CONFIGURING A NEW WLAN (GUEST)



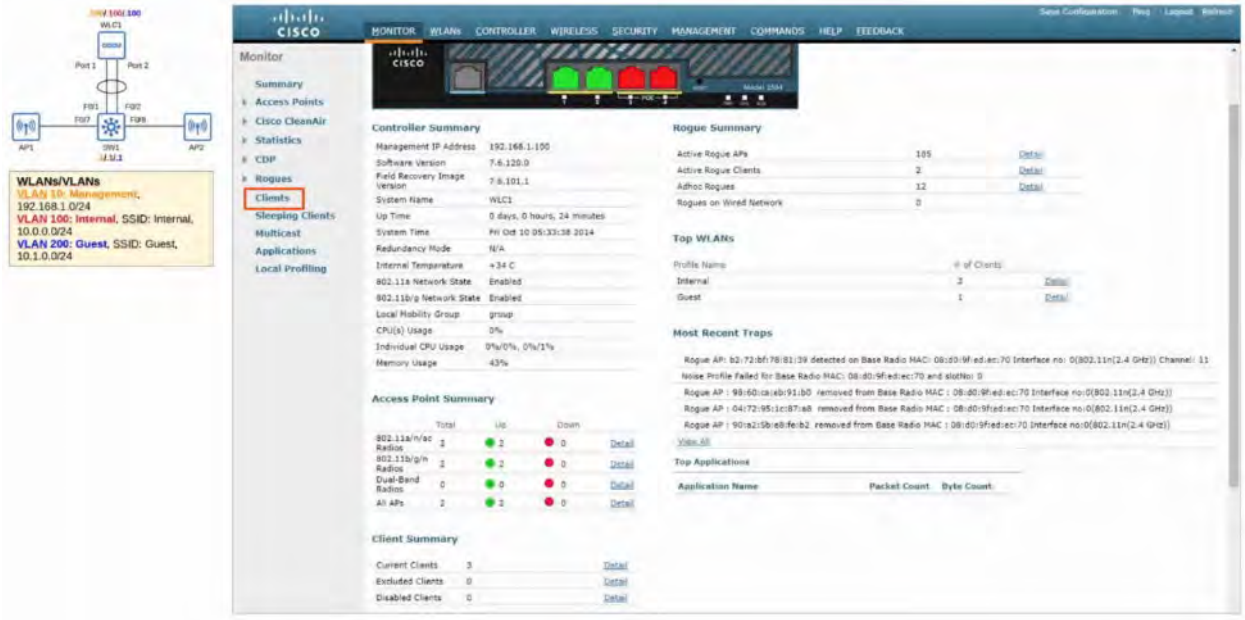
Change STATUS to “ENABLED” and INTERFACE GROUP to “GUEST”

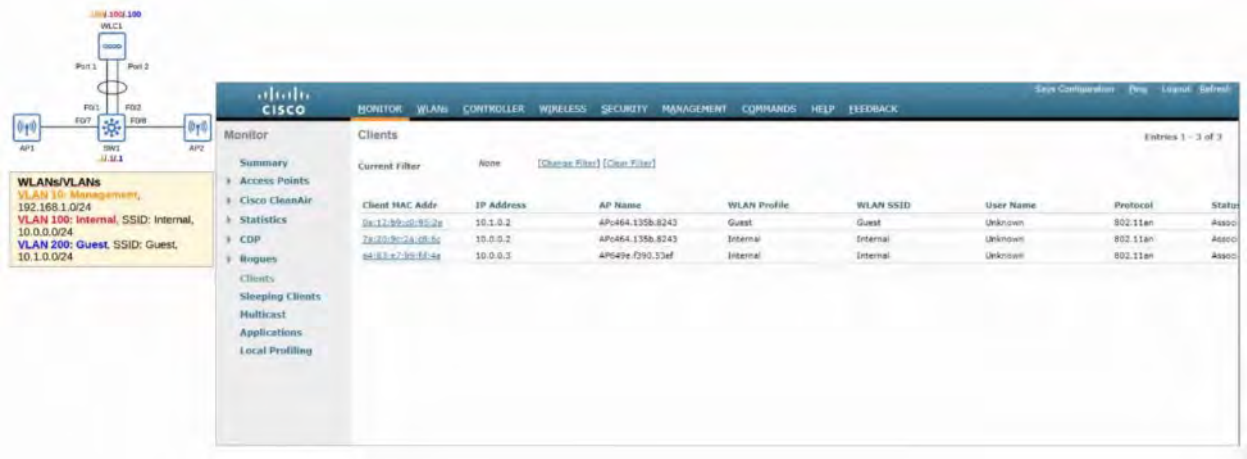


Now, we need to change the SECURITY POLICY to [WPA2][Auth(PSK)]
Returning to MONITORING, we can see the changes we made to the CONFIGURATION

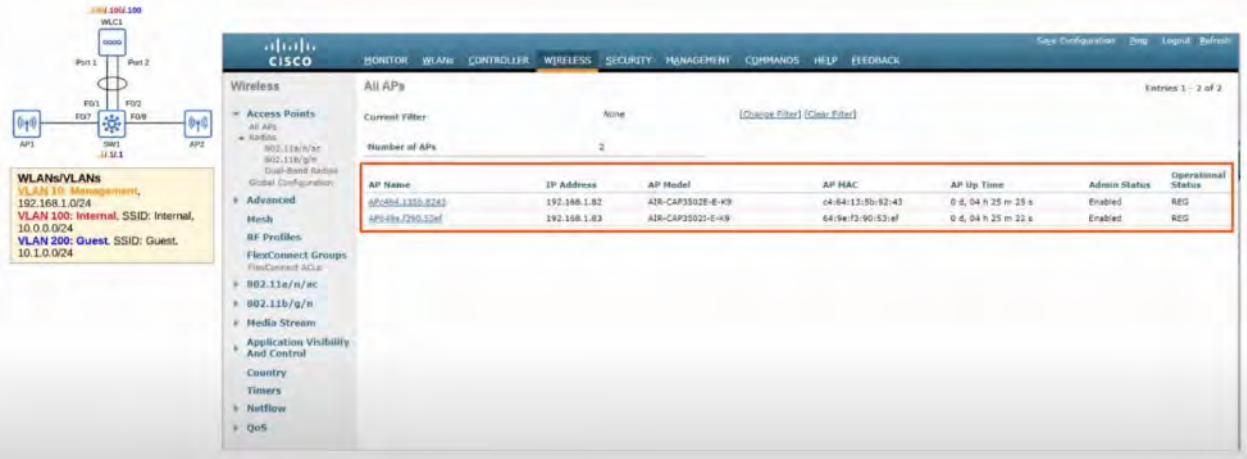


Current number of CLIENTS is now 0. By connecting to the WLANS, these numbers should change. To SEE a list of the CLIENTS connected, click the left-hand side “CLIENTS” tab

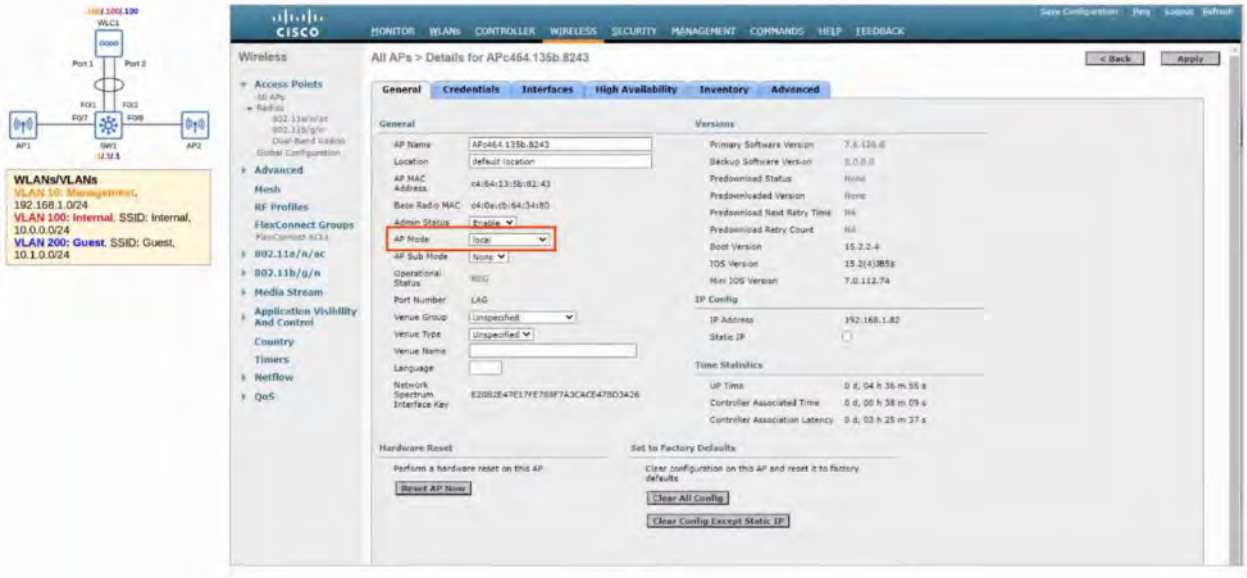




ADDTIONAL WLC FEATURES
WIRELESS tab showing a list of the APs currently in the NETWORK



Clicking on an AP shows information and configuration settings for it



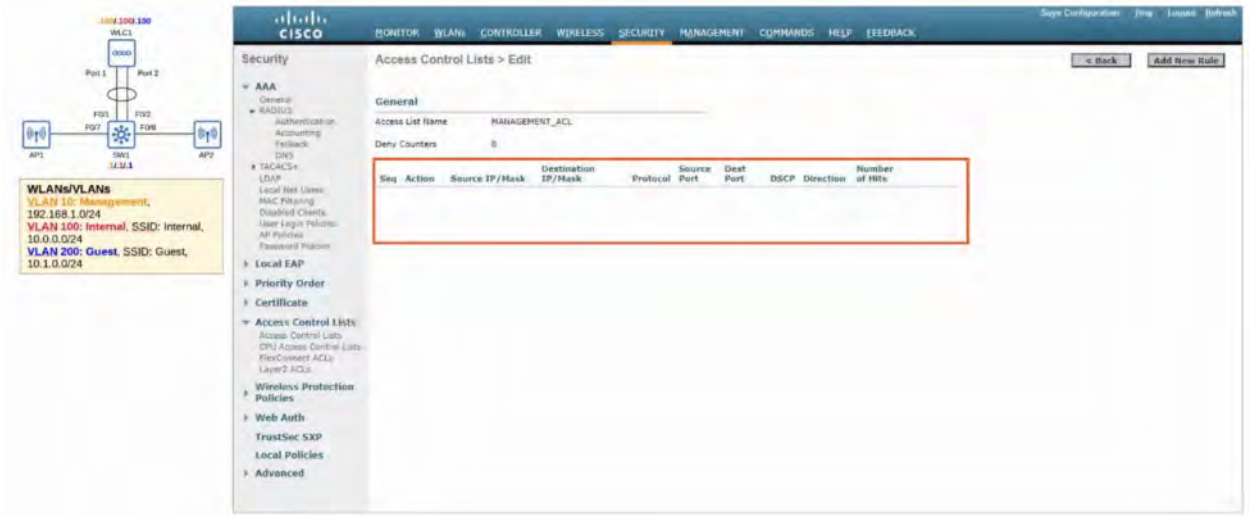
MANAGEMENT tab allows you change the ways you can MANAGE the WLC
Clicking “Mgmt Via Wireless” allows you change if you can access MANAGEMENT via WI-FI

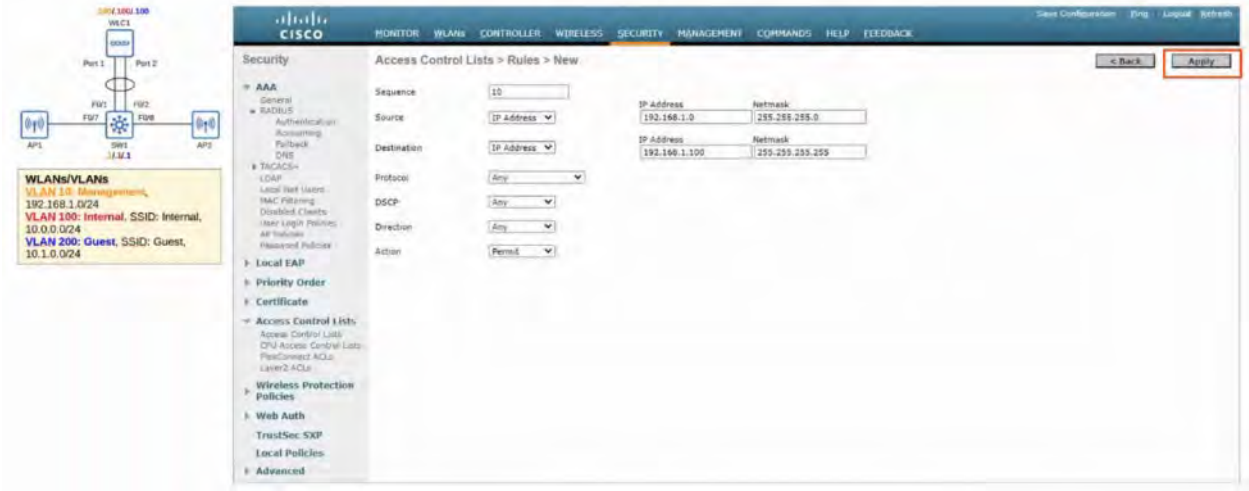

```
C:\Users\user>
C:\Users\user>telnet 192.168.1.100
Connecting To 192.168.1.100...Could not open connection to the host, on port 23: Connect failed

C:\Users\user>
```

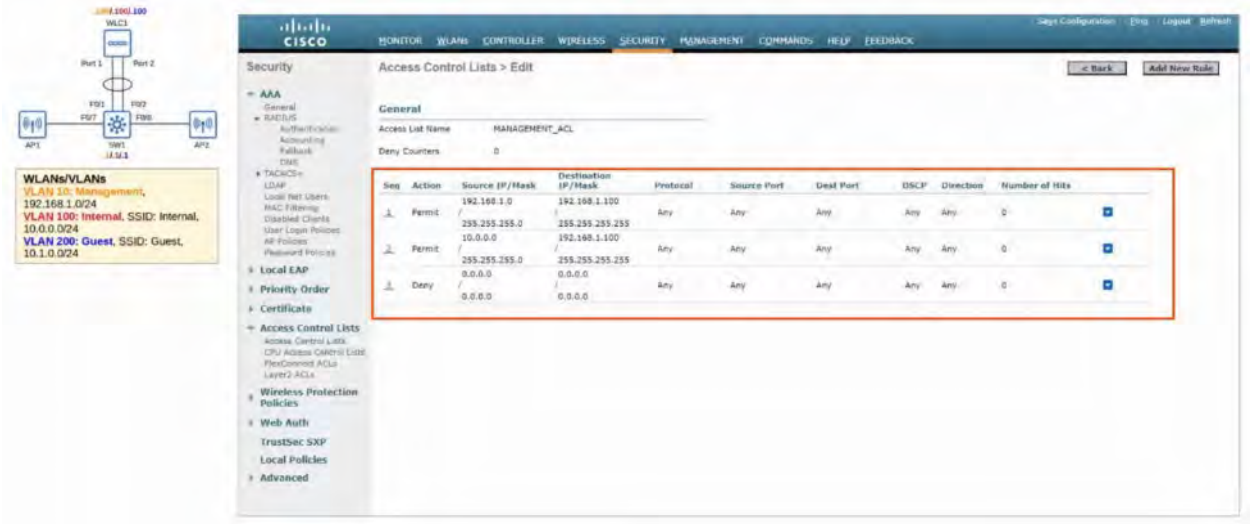
SECURITY tab can allow us to create ACCESS LISTS

First, NAME the ACL and what kind of IP ADDRESS it's for

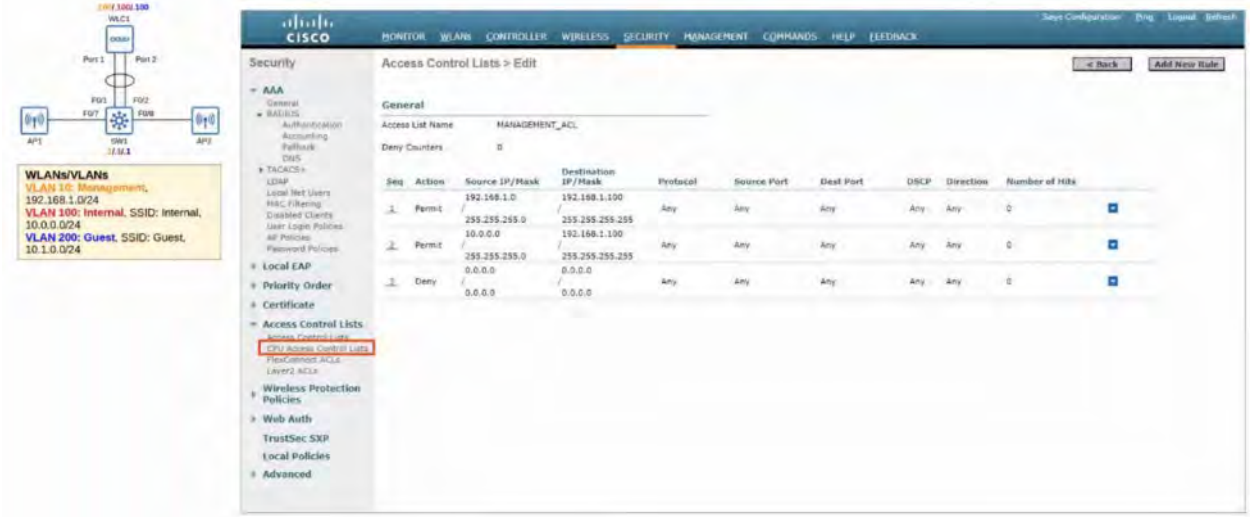
CLICK “Add New Rule” to specify the ACL Rules (What traffic can pass)



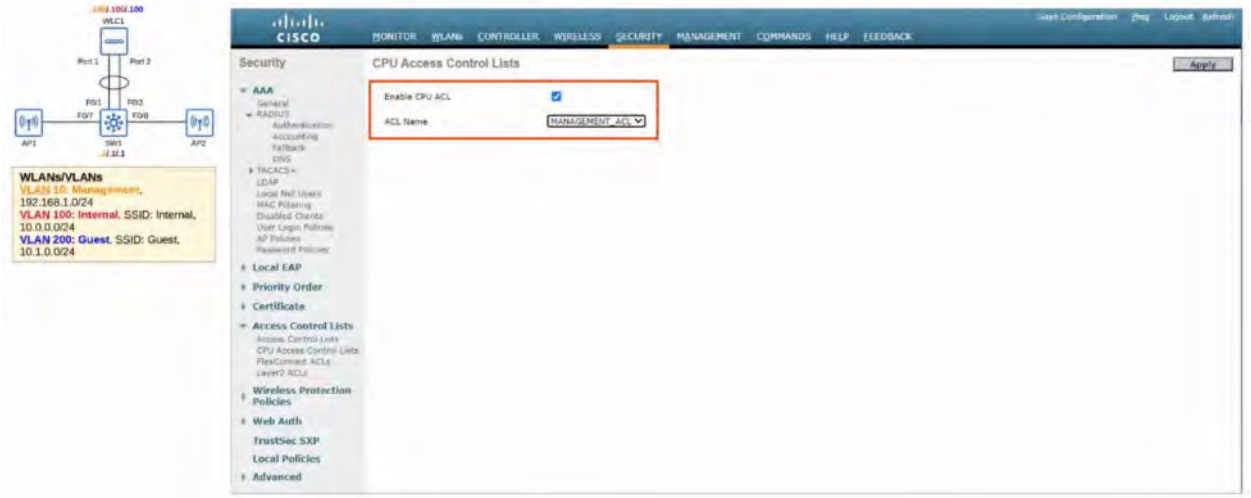




We now need to APPLY the ACL (just like applying it to an INTERFACE on a ROUTER)
Click “CPU ACL” from the left-hand menu



Select the new ACL from the pull-down list and then click “APPLY”



CPU ACLs are used to limit access to the CPU of the WLC. This limits which devices will be able to connect to the WLC via Telnet/SSH, HTTP/HTTPS, retrieve SNMP information from the WLC, etc.

59. INTRODUCTION TO NETWORK AUTOMATION

WHY NETWORK AUTOMATION

- Previous versions of the CCNA focused on the traditional model of managing / controlling networks
- The current version focuses on the traditional model as well, but CCNA candidates are expected to have a basic understanding of various topics related to network automation
- In the traditional model, engineers manage devices one at a time by connecting to their CLI via SSH

DOWNSIDES OF CONFIGURING DEVICES ONE-BY-ONE

- Typos and other small mistakes are common
- It is time-consuming and very inefficient in large-scale networks
- It is difficult to ensure that all devices ADHERE to the organization's STANDARD CONFIGURATION

BENEFITS OF NETWORK AUTOMATION

- Human Error (Typos, etc) is reduced
- Networks become much more scalable and implemented in a fraction of the time
    - o New deployments
    - o Network-wide changes
    - o Troubleshooting
- Network-wide policy compliance can be assured
    - o Standard configurations
    - o Software versioning
- The improved efficiency of network operations reduces the OP-EX (operating expenses) of the network. Each task requires fewer man-hours

There are various tools / methods that can be used to automate tasks in the network

- SDN (Software-Defined Networking)
- Ansible
- Puppet
- Python scripts
- etc…

LOGICAL "PLANES" OF NETWORK FUNCTIONS

What does a ROUTER do?

- It forwards messages between networks by examining information in the Layer 3 header
- It uses a routing protocol like OSPF to share route information with other routers and build a routing table
- It uses ARP to build an ARP table, mapping IP Addresses to MAC Addresses
- It uses Syslog to keep logs of events that occur
- and MUCH more…

What does a SWITCH do?

- It forwards messages within a LAN by examining information in the Layer 2 header
- It uses STP to ensure there are no Layer 2 loops in the network
- It builds a MAC address table by examining the Source MAC address of frames
- It uses Syslog to keep logs of events that occur
- It allows a user to connect to it via SSH and manage it

The various functions of network devices can be logically divided up (categorized) into *PLANES*

- DATA PLANE
- CONTROL PLANE
- MANAGEMENT PLANE

- The operations of the MANAGEMENT PLANE and the CONTROL PLANE are usually managed by the CPU
- However, this is not desirable for DATA PLANE operations because CPU processing is slow (relatively speaking)
- Instead, a specialized hardware ASIC (Application-Specific Integrated Circuit) is used.
    - ASICs are chips built for a specific purpose
- Using a SWITCH, as an example:
    - When a FRAME is received, the ASIC (not the CPU) is responsible for the switching logic
    - The MAC Address table is stored in a kind of memory called TCAM (Ternary Content-Addressable Memory)
        - Another common name for the MAC Address table is CAM TABLE
    - The ASIC feeds the DESTINATION MAC address of the FRAME into the TCAM which returns the matching MAC Address table entry
    - The FRAME is then forwarded out of the appropriate DEVICE
- Modern ROUTERS also use a similar hardware DATA PLANE: An ASIC designed for forwarding logic, and tables store in TCAM

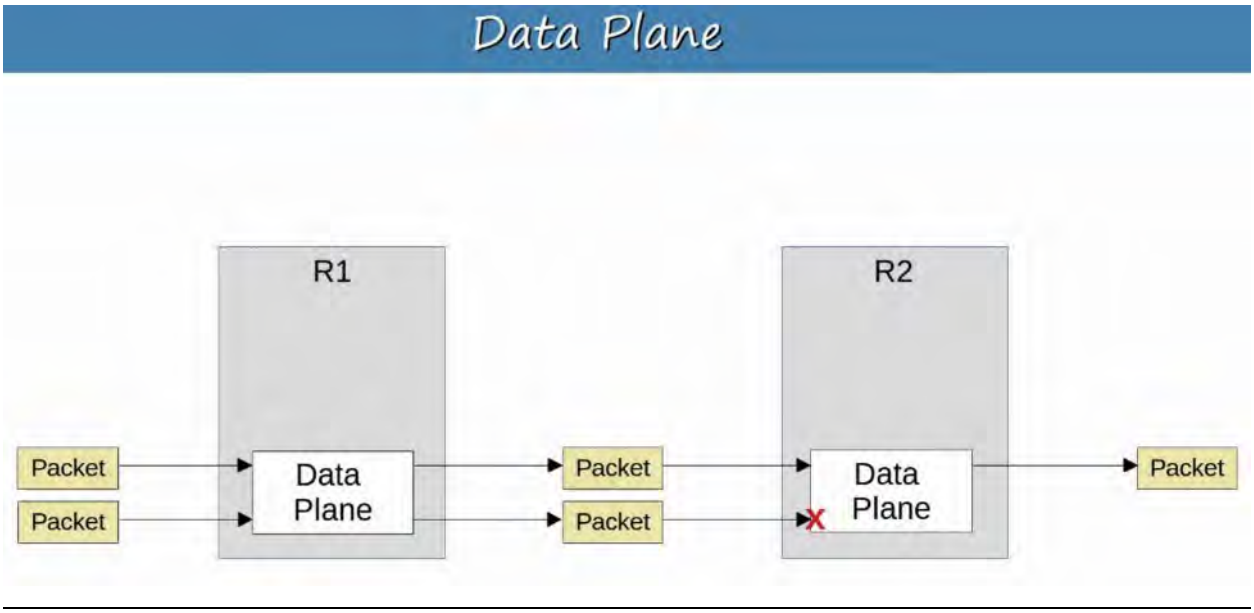
A SIMPLE SUMMARY:

- When a DEVICE receives CONTROL / MANAGEMENT traffic (destined for itself), it will be processed in the CPU
- When a DEVICE receives DATA traffic which should pass through the DEVICE, it is processed by the ASIC for maximum speed

---

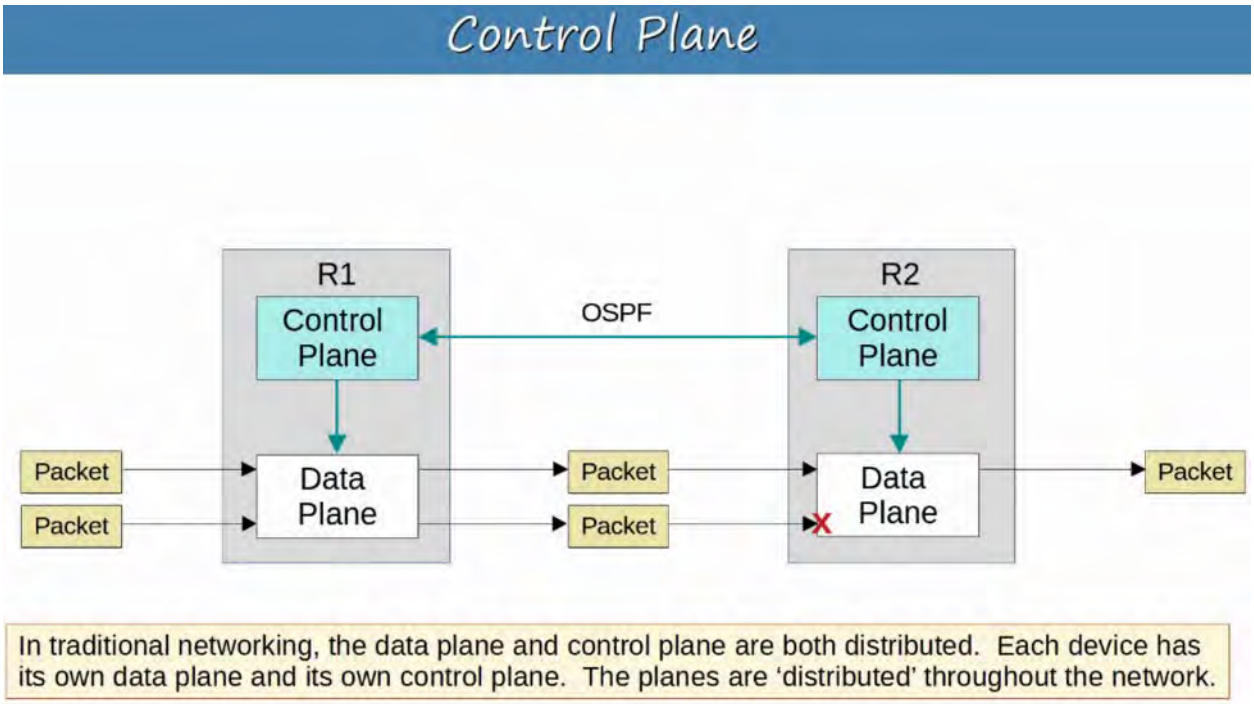
DATA PLANE

- All tasks involved in forwarding USER DATA / TRAFFIC from one INTERFACE to another are part of the DATA PLANE
- A ROUTER receives a message, looks for the most specific matching ROUTER in its ROUTING TABLE, and forwards it out of the appropriate INTERFACE to the next hop
    - It also de-encapsulates the original LAYER 2 header, and re-encapsulates with a new header destined for the next hop's MAC address
- A SWITCH receives a message, looks at the DESTINATION MAC Address, and forwards it out of the appropriate INTERFACE (or FLOODS it)
    - This includes functions like adding / removing 802.1q VLAN tags
- NAT (changing the SRC / DST addresses before forwarding) is part of the DATA PLANE
- Deciding to forward / discard messages due to ACL's, port-security, etc. is part of the DATA PLANE
- The DATA PLANE is also called the 'FORWARDING PLANE'



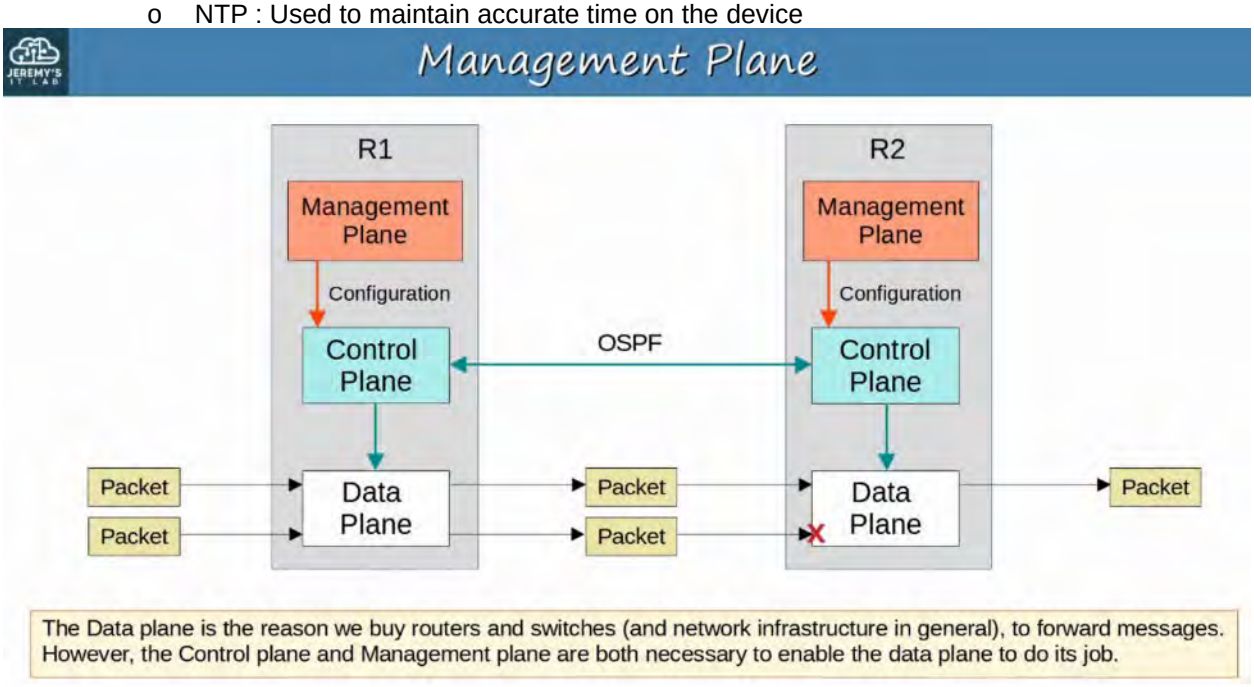
CONTROL PLANE

- How does a DEVICE's DATA PLANE make its forwarding decisions?
  - ROUTING TABLE
  - MAC ADDRESS table
  - ARP table
  - STP
  - etc…
- Functions that build THESE tables (and other functions that influence the DATA PLANE) are part of the CONTROL PLANE
- The CONTROL PLANE *controls* what the DATA PLANE does, for example by building the ROUTER's ROUTING TABLE
- The CONTROL PLANE performs *overhead* work
  - OSPF itself doesn't forward user data packets, but it informs the DATA PLANE about HOW packets should be forwarded
  - STP itself isn't directly involved in the process of forwarding FRAMES, but it informs the DATA PLANE about which INTERFACES should and shouldn't be used to forward FRAMES
  - ARP messages aren't user data but they are used to build an ARP TABLE which is used in the process of forwarding data



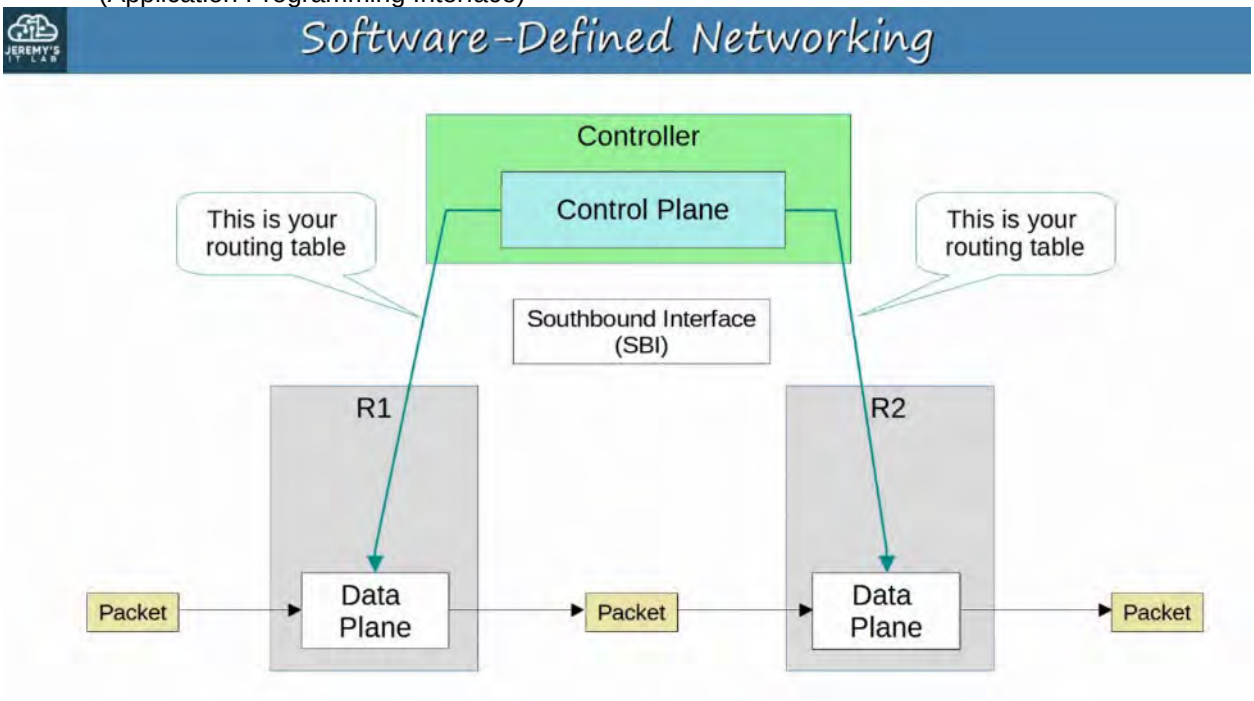
MANAGEMENT PLANE

- Like the CONTROL PLANE, the MANAGEMENT PLANE performs overhead work
  - However, the MANAGEMENT PLANE doesn't directly affect the forwarding of messages in the DATA PLANE
- The MANAGMENT PLANE consists of PROTOCOLS that are used to manage devices
  - SSH / TELNET : Used to connect to the CLI of a DEVICE to configure / manage it
  - SYSLOG : Used to keep logs of events that occur on the device
  - SNMP : Used to monitor the operations of the device
  - NTP : Used to maintain accurate time on the device



SOFTWARE-DEFINED NETWORKING (SDN)

- SOFTWARE-DEFINED NETWORKING (SDN) is an approach to networking that centralizes the CONTROL PLANE into an application called a *CONTROLLER*
- SDN is also called SOFTWARE-DEFINED-ARCHITECTURE (SDA) or CONTROLLER-BASED NETWORKING
- Traditional CONTROL PLANES use a distributed architecture
    - For example:
        - Each ROUTER in the NETWORK runs OSPF and the ROUTERS share routing information and then calculate their preferred routes to each destination
- An SDN CONTROLLER centralized CONTROL PLANE functions like calculation routes
    - That is just an example and how much of the CONTROL PLANE is centralized varies greatly
- The CONTROLLER can interact programmatically with the NETWORK DEVICE using APIs (Application Programming Interface)



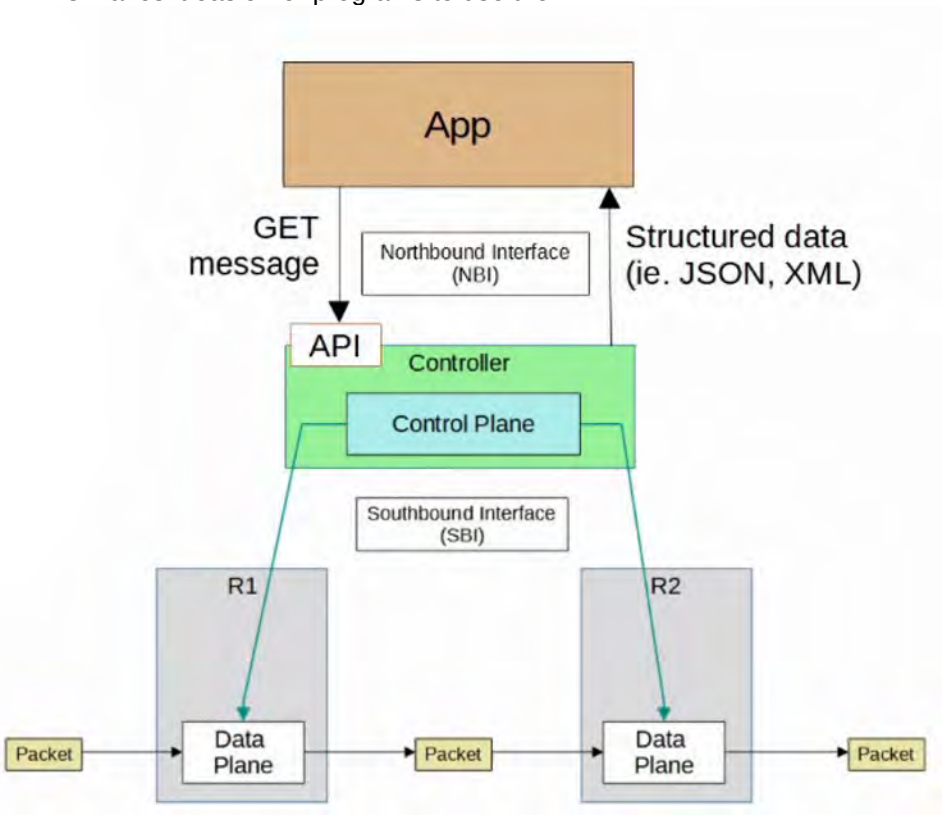
SOUTHBOUND INTERFACE (SBI)

- The SBI is used for communications between the CONTROLLER and the NETWORK DEVICES it controls
- It typically consists of a COMMUNICATION PROTOCOL and API (Application Programming Interface)
- APIs facilitate data exchanges between programs
    - DATA is exchanged between the CONTROLLER and the NETWORK DEVICES
    - An API on the NETWORK DEVICES allows the CONTROLLER to access information on the DEVICES, control their DATA PLANE TABLES, etc.
- Some examples of SBIs :
    - OpenFlow
    - Cisco OpFlex
    - Cisco OnePK (Open Network Environment Platform Kit)
    - NETCONF

NORTHBOUND INTERFACE (NBI)

- Using the SBI, the CONTROLLER communicates with the managed DEVICES and gathers information about them:

    - The DEVICES in the NETWORK
    - The TOPOLOGY (how the DEVICES are connected together)
    - The available INTERFACES on each DEVICE
    - Their CONFIGURATIONS
- The NORTHBOUND INTERFACE (NBI) is what allows us to:
    - Interact with the CONTROLLER
    - Access the DATA it gathers about the NETWORK
    - Program the NETWORK
    - Make changes to the NETWORK via the SBI
- A REST API (Representational State Transfer) is used on the controller as an interface for APPS to interact with it
- OSGi (Java Open Services Gateway Initiative) - Java based NBI API
- DATA is sent in a structured (*serialized*) format such as JSON or XML
    - This makes it easier for programs to use the DATA



AUTOMATION IN TRADITIONAL NETWORKS VS SDN

- Networking tasks can be automated in traditional NETWORK architectures too:
    - SCRIPTS can be written (ie: using Python) to push commands to many DEVICES at once
    - Python with good use of REGULAR EXPRESSIONS can parse through “show” commands to gather information about network devices
- However, the robust and centralized DATA collected by SDN CONTROLLERS greatly facilitates these functions
    - The CONTROLLER collects information about all DEVICES in the NETWORK
    - NORTHBOUND APIs allow APPS to access information in a format that is easy for programs to understand (ie: JSON and XML)
    - The centralized DATA facilitates network-wide analytics
- SDN Tools can provide the benefits of automation without the requirement of third-party scripts and apps.

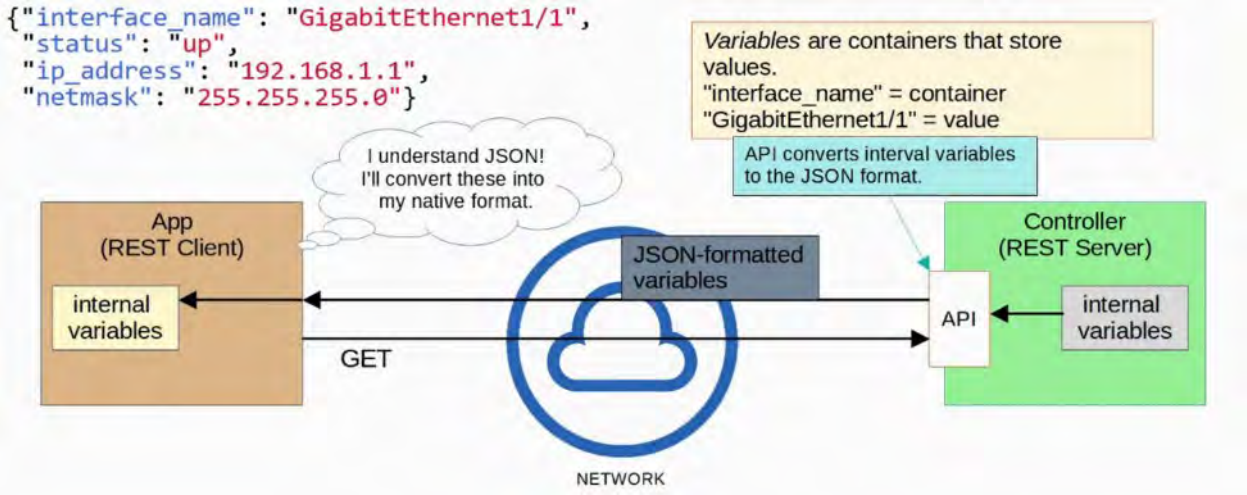
- You don’t need expertise in automation to make use of SDN Tools
- However, APIs allow third-party applications to interact with the CONTROLLER, which can be very powerful

💡 Although SDN and automation aren’t the same thing, the SDN architecture greatly facilitates the automation of various tasks in the network via the SDN CONTROLLER and APIs

60. JSON, XML, AND YAML

DATA SERIALIZATION

- DATA SERIALIZATION is the process of converting DATA into a standardized format/structure that can be stored (in a file) or transmitted (over a network) and reconstructed later (ie: by a different application)
    - This allows the DATA to be communicated between applications in a way both APPLICATIONS understand.
- DATA SERIALIZATION languages allow us to represent *variables* with text



JSON (JAVASCRIPT OBJECT NOTATION)

- JSON (JAVASCRIPT OBJECT NOTATION) **is an open standard FILE FORMAT and DATA INTERCHANGE FORMAT that uses human-readable text to store and transmit data objects
- It is standardized in RFC 8259 (https://datatracker.ietf.org/doc/html/rfc8259)
- It was derived from JavaScript, but it is language-independent and many modern programming languages are able to generate and read JSON data
    - REST APIs often use JSON
- *Whitespace* is insignificant
- JSON can represent FOUR “primitive” DATA Types:
    - String
    - Number
    - Boolean
    - Null
- JSON also has TWO “structured” DATA Types:
    - Object
    - Array

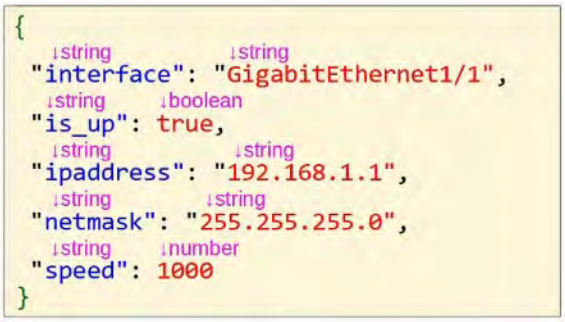
JSON PRIMITIVE DATA TYPES:

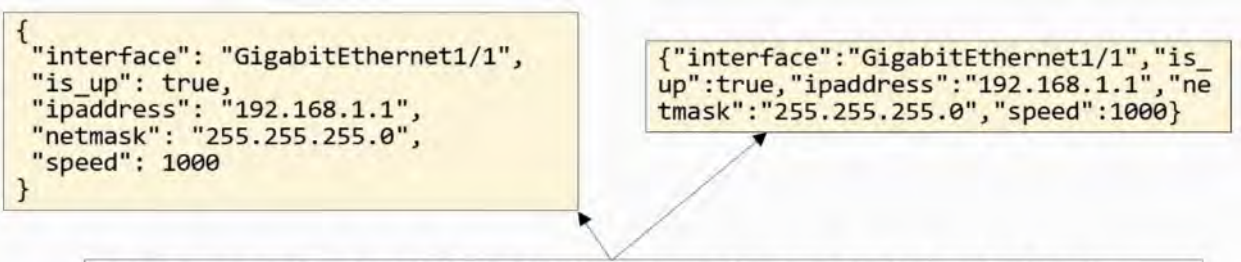
- A STRING is a text value. It is surrounded by double quotes “ “
    - “Hello”
    - “Five”
    - “5”
- A NUMBER is a numeric value. It is NOT surrounded by quotes
    - 5
    - 100
- A BOOLEAN is a DATA Type that has only TWO possible values, not surrounded by quotes
    - true
    - false

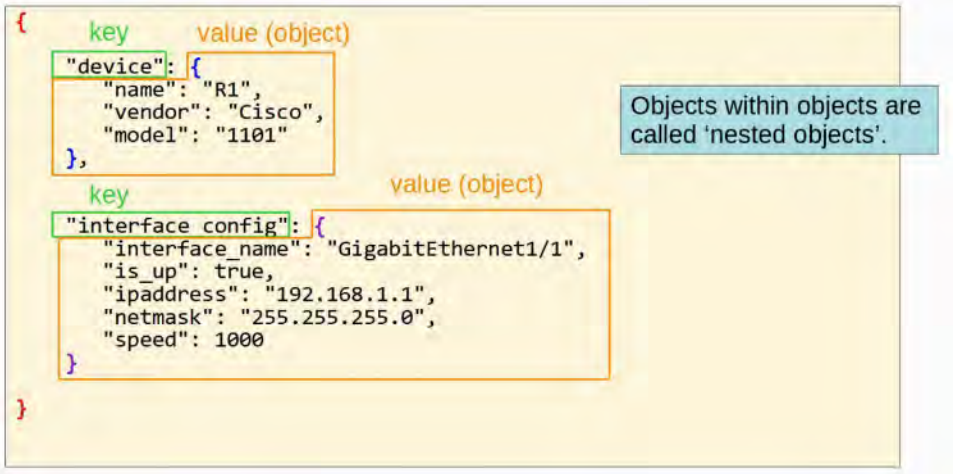
- A NULL value represents the intentional absence of any object value. It is not surrounded by quotes
    - null

JSON STRUCTURED DATA TYPES:

- An OBJECT is an unordered list of *key-value pairs* (variables)
    - Sometimes called a DICTIONARY
    - OBJECTS are surrounded by curly brackets {}
    - The *key* is a STRING
    - The *value* is any valid JSON DATA Type (string, number, boolean, null, object, array)
    - The *key* and *value* are separated by a colon :
    - If there are multiple key-value pairs, each pair is separated by a comma







- An ARRAY is a series of *values* separated by commas
    - Not *key-value pairs*
    - The values do NOT have to be the same DATA Type

```
{
    "interfaces": [
        "GigabitEthernet1/1",
        "GigabitEthernet1/2",
        "GigabitEthernet1/3"
    ],

    "random_values": [
        "Hi",
        5
    ]
}
```

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         192.168.1.1     YES manual up                    up
GigabitEthernet0/1         unassigned      YES unset  administratively down down
```

```
{
    "ip_interfaces": [
            {
            "Interface":"GigabitEthernet0/0",
            "IP-Address": "192.168.1.1",
            "OK?": "YES",
            "Method": "manual",
            "Status": "up",
            "Protocol": "up"
            },
            {
            "Interface":"GigabitEthernet0/1",
            "IP-Address": "unassigned",
            "OK?": "YES",
            "Method": "unset",
            "Status": "administratively down",
            "Protocol": "down"
            }
    ]
}
```

XML (EXTENSIBLE MARKUP LANGUAGE)

- XML (EXTENSIBLE MARKUP LANGUAGE) was developed as a MARKUP language, but is now used as a general data serialization language
    - Markup languages (ie: HTML) are used to format text (font, size, color, headings, etc)
    - XML is generally less human-readable than JSON
    - Whitespace is insignificant
    - Often used by REST APIs
    - value (similar to HTML)

```
R1#show ip interface brief | format
<?xml version="1.0" encoding="UTF-8"?>
  <ShowIpInterfaceBrief xmlns="ODM://built-in//show_ip_interface_brief">
    <SpecVersion>built-in</SpecVersion>
    <IPInterfaces>
      <entry>
        <Interface>GigabitEthernet0/0</Interface>
        <IP-Address>192.168.1.1</IP-Address>
        <OK>YES</OK>
        <Method>manual</Method>
        <Status>up</Status>
        <Protocol>up</Protocol>
      </entry>
      <entry>
        <Interface>GigabitEthernet0/1</Interface>
        <OK>YES</OK>
        <Method>unset</Method>
        <Status>administratively down</Status>
        <Protocol>down</Protocol>
      </entry>
    </IPInterfaces>
  </ShowIpInterfaceBrief>
```

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         192.168.1.1     YES manual up                    up
GigabitEthernet0/1         unassigned      YES unset  administratively down down


R1#show ip interface brief | format
<?xml version="1.0" encoding="UTF-8"?>
  <ShowIpInterfaceBrief xmlns="ODM://built-in//show_ip_interface_brief">
    <SpecVersion>built-in</SpecVersion>
    <IPInterfaces>
      <entry>
        <Interface>GigabitEthernet0/0</Interface>
        <IP-Address>192.168.1.1</IP-Address>
        <OK>YES</OK>
        <Method>manual</Method>
        <Status>up</Status>
        <Protocol>up</Protocol>
      </entry>
      <entry>
        <Interface>GigabitEthernet0/1</Interface>
        <OK>YES</OK>
        <Method>unset</Method>
        <Status>administratively down</Status>
        <Protocol>down</Protocol>
      </entry>
    </IPInterfaces>
  </ShowIpInterfaceBrief>
```

YAML (YAML AIN'T MARKUP LANGUAGE)

- YAML originally meant *YET ANOTHER MARKUP LANGUAGE* but to distinguish its purpose as a data-serialization language rather than a markup language, it was repurposed to *YAML AINT MARKUP LANGUAGE*
- YAML is used by the network automation tool ANSIBLE (covered later in the course)
- YAML is VERY Human-Readable
- Whitespace **is significant** (unlike JSON and XML)
    - Indentation is very important
- YAML files start with - - - (three dashes)
-
    - is used to indicate a list
- Keys and Values are represented as key : value

```
---
ip_interfaces:
- Interface: GigabitEthernet0/0
  IP-Address: 192.168.1.1
  OK?: 'YES'
  Method: manual
  Status: up
  Protocol: up
- Interface: GigabitEthernet0/1
  IP-Address: unassigned
  OK?: 'YES'
  Method: unset
  Status: administratively down
  Protocol: down
```

COMPARISON BETWEEN JSON and YAML using the same DATA

JSON

```
{
    "ip_interfaces": [
            {
            "Interface":"GigabitEthernet0/0",
            "IP-Address": "192.168.1.1",
            "OK?": "YES",
            "Method": "manual",
            "Status": "up",
            "Protocol": "up"
            },
            {
            "Interface":"GigabitEthernet0/1",
            "IP-Address": "unassigned",
            "OK?": "YES",
            "Method": "unset",
            "Status": "administratively down",
            "Protocol": "down"
            }
    ]
}
```

YAML

```
---
ip_interfaces:
- Interface: GigabitEthernet0/0
  IP-Address: 192.168.1.1
  OK?: 'YES'
  Method: manual
  Status: up
  Protocol: up
- Interface: GigabitEthernet0/1
  IP-Address: unassigned
  OK?: 'YES'
  Method: unset
  Status: administratively down
  Protocol: down
```

61. REST APIS

API REVIEW

- An API (Application Programming Interface) is a software interface that allows two applications to communicate with each other.
- APIs are essential not just for network automation but for all kinds of applications
- In SDN Architecture, APIs are use to communicate between apps and the SDN controller (via the NBI) and between the SDN controller and the network devices (via the SBI)
- The NBI typically uses REST APIs
- NETCONF and RESTCONF are popular Southbound APIs

CRUD OPERATIONS AND HTTP VERBS

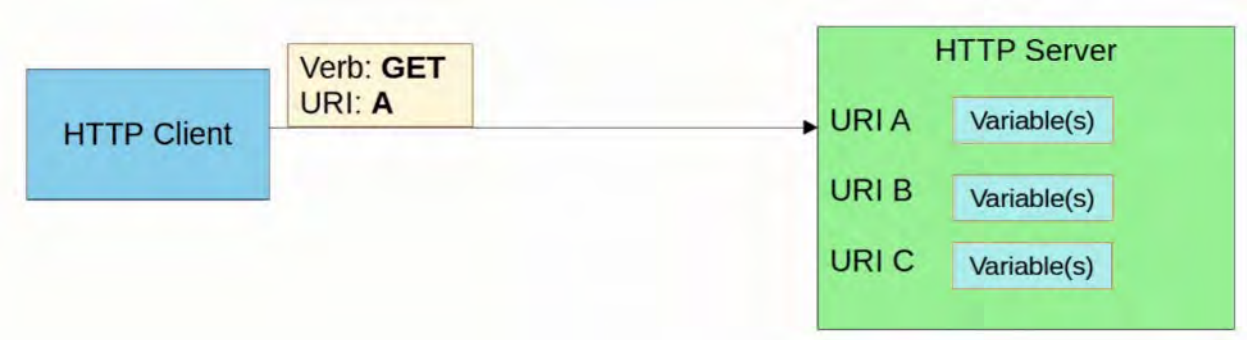
- CRUD (CREATE, READ, UPDATE, DELETE) refers to the operations we perform using REST APIs
- CREATE :
  - Used to CREATE new variables and set their initial values
    - Example: create a variable “ip_address” and set the value to “10.1.1.1”
- READ :
  - Used to READ the value of a variable
    - Example: Read the value of variable “ip_address” (”10.1.1.1”)
- UPDATE :
  - Used to CHANGE / UPDATE the value of a variable
    - Example: Change the value of “ip_address” from “10.1.1.1” to “10.2.3.4”
- DELETE :
  - Used to DELETE variables
    - Example: Delete variable “ip_address”
- HTTP uses *verbs* (aka. methods) that map to these CRUD operations
- REST APIs typically use HTTP

HTTP Verbs

Purpose	CRUD Operation	HTTP Verb
Create new variable	**Create**	**POST**
Retrieve value of variable	**Read**	**GET**
Change the value of variable	**Update**	**PUT, PATCH**
Delete variable	**Delete**	**DELETE**

HTTP REQUEST :

- When an HTTP client sends a request to an HTTP server, the HTTP header includes information like this:
  - An HTTP Verb (ie: GET)
  - A URI (Uniform Resource Identifier) indicating the resource it is trying to access



An example of a URI (demonstrated later)

- The HTTP request can include additional headers which pass additional information to the server. Check the list at https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

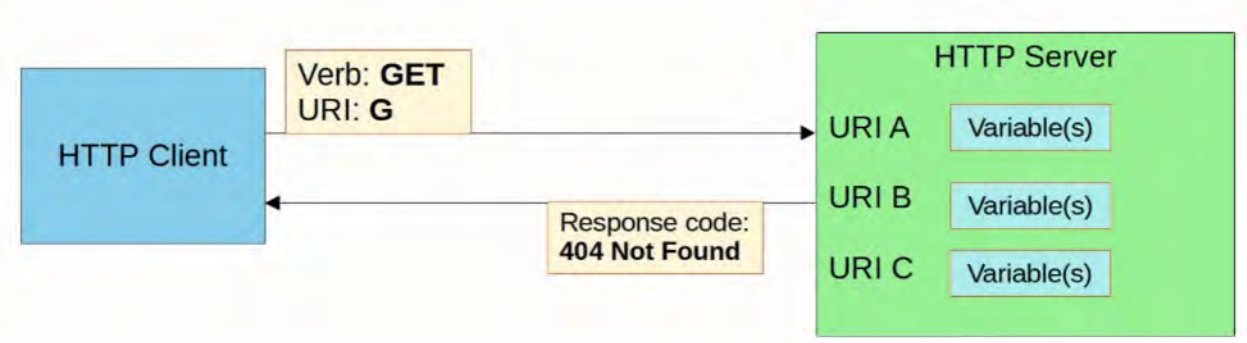
- An example would be an ACCEPT header, which informs the server about the types(s) of data that can be sent back to the client.
    - Example: **Accept: application/json** or **Accept: application/xml**
- You can also view standard HTTP header fields with some examples at https://en.wikipedia.org/wiki/List_of_HTTP_header_fields
- When a REST client makes an API call (request) to a REST server, it will send an HTTP request like the one above

💡 REST APIs do NOT have to use HTTP for communication, although HTTP is the most common choice

---

HTTP RESPONSE :

- The server's response will include a STATUS CODE indicating if the request succeeded or failed, as well as other details
- The FIRST digit indicates the class of the response:
    - 1xx : Informational - request was received, continuing process
    - 2xx : Successful - request was successfully received, understood, and accepted
    - 3xx : Redirection - further action needs to be taken in order to complete the request
    - 4xx : Client Error - request contains bad syntax or cannot be fulfilled
    - 5xx : Server Error - server failed to fulfill an apparently valid request



Examples of each HTTP Response class:

- 1xx Informational

    - 102 Processing indicates that the server received the request and is processing it but the response is not available yet
- 2xx Successful
    - 200 OK **indicates that the request succeeded
    - 201 Created indicates the request succeeded and a new resource was created
- 3xx Redirection
    - 301 Moved Permanently indicates that the request resource has been moved and the server indicates its new location
- 4xx Client Error
    - 403 Unauthorized means the client must authenticate to get a response
    - 404 Not Found means the requested resource was not found
- 5xx Server Error
    - 500 Internal Server Error means the server encountered something unexpected that it doesn't know how to handle

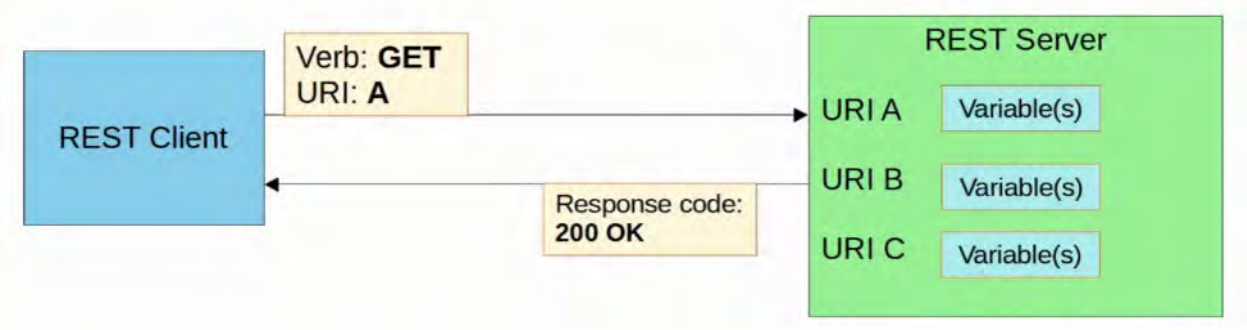
---

REST APIs

- REST stands for Representational State Transfer
- REST APIs are also know as REST-based APIs or RESTful APIs
    - REST isn't a specific API. Instead it describes a set of rules about how the API should work
- The SIX constraints of RESTful architecture are:
    - Stateless
    - Layered system
    - Uniform Interface
    - Client-Server
    - Cacheable or non-cacheable
    - Code-on-Demand (optional)
- For applications to communicate over a network, networking protocols must be used to facilitate those communications
    - For REST APIs, HTTP(S) is the most common choice

REST: Client-Server

- REST APIs use a client-server architecture
- The client uses API calls (HTTP requests) to access the resources on the server
- The separation between the client and server means they can both change and evolve independently of each other
    - When the client application changes or the server application changes, the interface between them must not break



REST: Stateless

- REST APIs exchanges are STATELESS
- This means that each API exchange is a separate event, independent of all past exchanges between the client and server

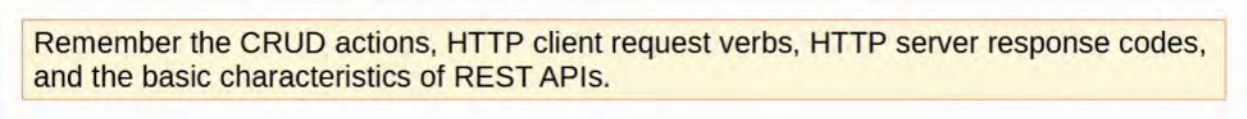
    - The server does not store information about previous requests from the client to determine how it should respond to new requests
- If authentication is required, this means that the client must authenticate with the server for each request it makes
- TCP is an example of a STATEFUL protocol
- UDP is an example of STATELESS protocol

** Although REST APIs use HTTP, which uses TCP (STATEFUL) as it's LAYER 4 protocol, HTTP and REST APIs themselves aren't STATEFUL. The functions of each layer are separate !

REST: Cacheable or Non-Cacheable

- REST APIs must support caching of data
- *Caching* refers to storing data for future use
    - Example :
        - Your computer might cache many elements of a web page so it doesn't have to retrieve the entire page every time you visit. This improves performance for the client and reduces load on the server
- Not all resources have to be cacheable but cacheable resources MUST be declared as cacheable

FOR THE CCNA



REST API CALLS USING CISCO DEVNET

- "Cisco DevNet is Cisco's developer program to help developers and IT professionals who want to write applications and develop integrations with Cisco products, platforms, and API's"
- DevNet offers lots of free resources such as courses, tutorials, labs, sandboxes, documentation, etc to learn about AUTOMATION and develop your skills
- There is also a DevNet certification track that you can pursue if you are interested in AUTOMATION
- We will use their Cisco DNA Center Sandbox to send a REST API call using Postman
    - DNA Center is one of Cisco's SDN Controllers (covered in more detail later)
    - Postman is a platform for building an using APIs

TO START:

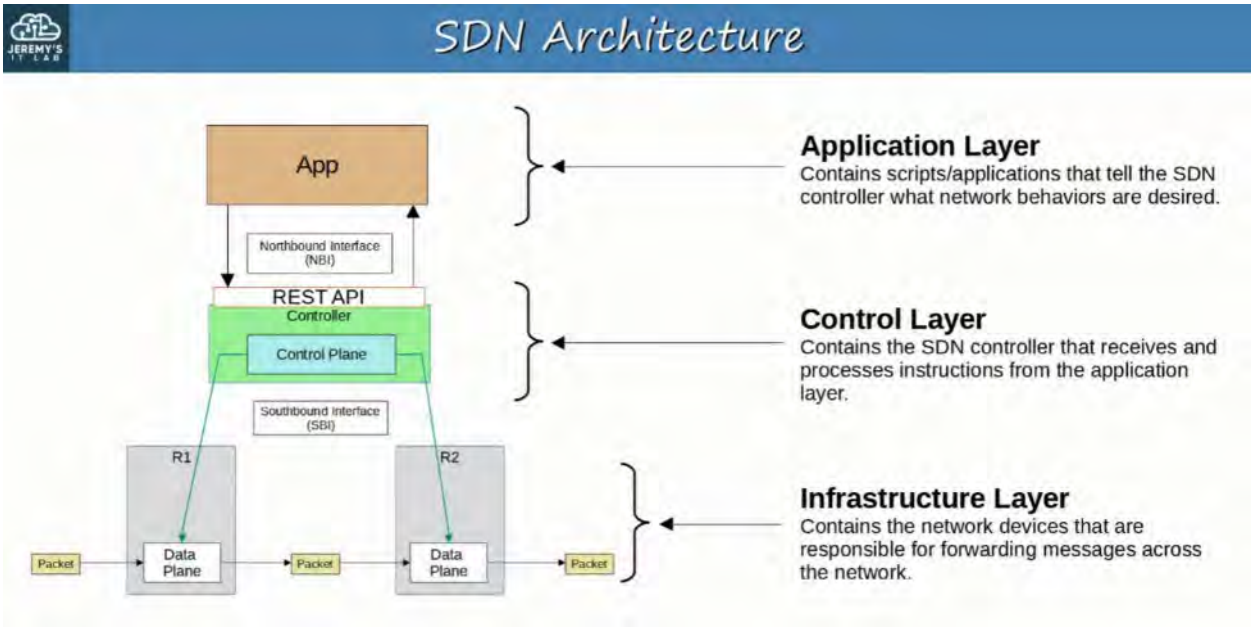
- Make an account on developer.cisco.com (Used my NetAcademy login)
- Make an accounts on postman.com and download the desktop app (https://www.postman.com/downloads) - Used my gmail.com account

62. SOFTWARE DEFINED NETWORKING (SDN)

SD REVIEW

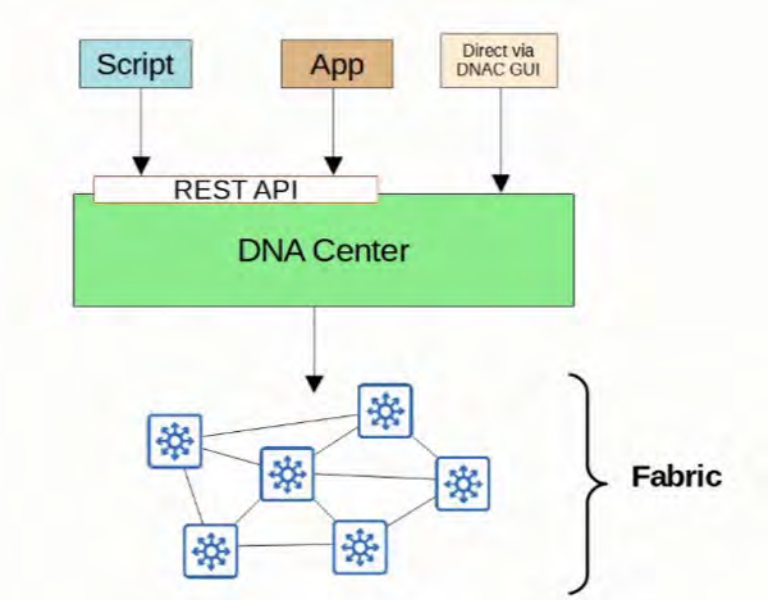
- SOFTWARE DEFINED NETWORKING (SDN) is an approach to networking that centralizes the control plane into an application called a *controller*
- Traditional control planes use a distributed architecture
- A SDN controller centralizes control plane functions like calculating routes
- The controller can interact programmatically with the network devices using APIs
- The SBI (South Bound Interface) is used for communications between the controller and the network device it controls
- The NBI (North Bound Interface) is what allows us to interact with the controller with our scripts and applications

SDN ARCHITECTURE

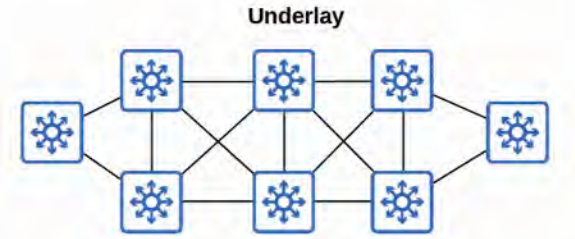


CISCO SD-ACCESS

- Cisco SD-ACCESS is Cisco's SDN solution for automating campus LANs
  - ACI (Application Centric Infrastructure) is their SDN solution for automating data center networks
  - SD-WAN is their SDN solution for automating WANs
- Cisco DNA (Digital Network Architecture) Center is the controller at the center of SD-Access

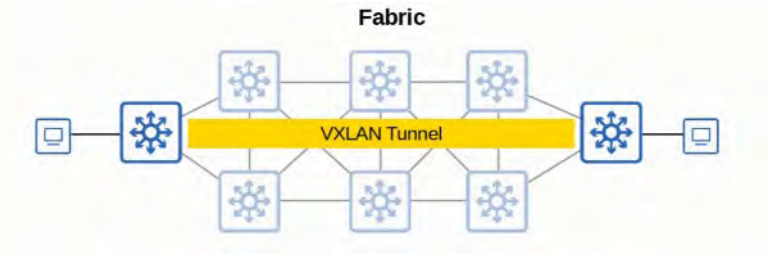


- The UNDERLAY is the underlying physical network of devices and connections (including wired and wireless) which provide IP connectivity (ie: using IS-IS)
    - Multilayer Switches and their connections



- The OVERLAY is the virtual network built on top of the physical underlay network

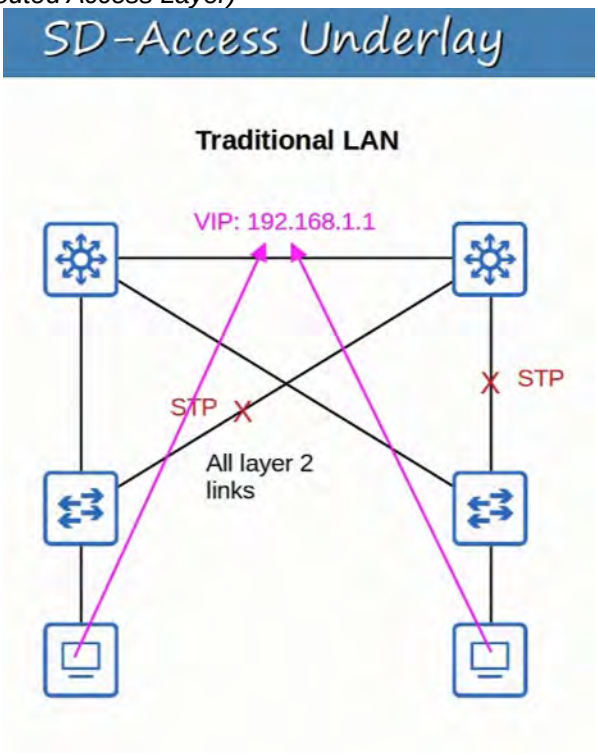
- The FABRIC is the combination of the OVERLAY and UNDERLAY; the physical and virtual network as a whole

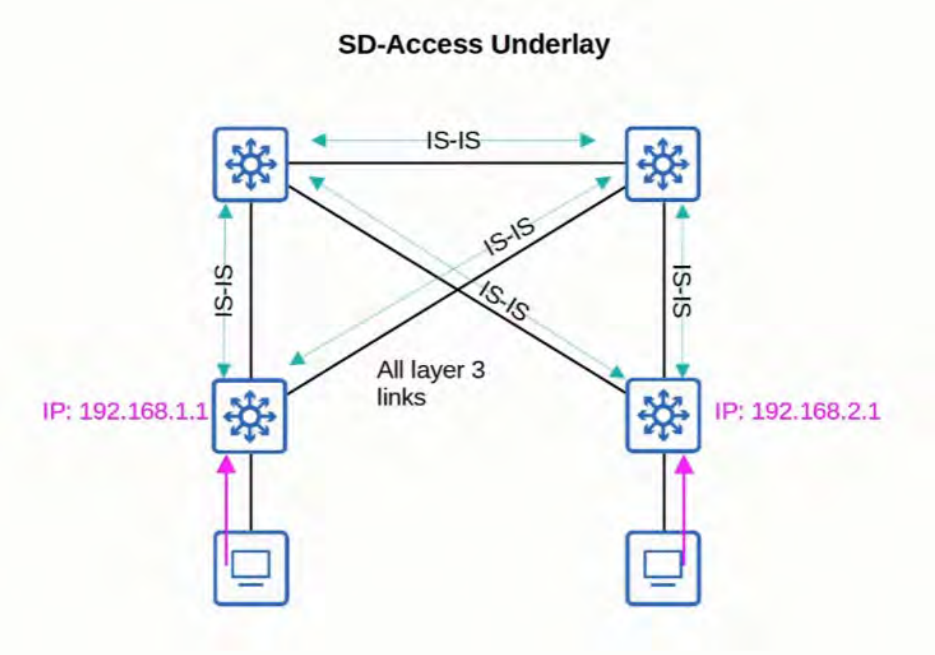


SD-ACCESS UNDERLAY

- The UNDERLAY's purpose is to support the VXLAN tunnels of the OVERLAY
- There are THREE different ROLES for switches in SD-ACCESS:
    - EDGE NODES : Connect to End HOSTS

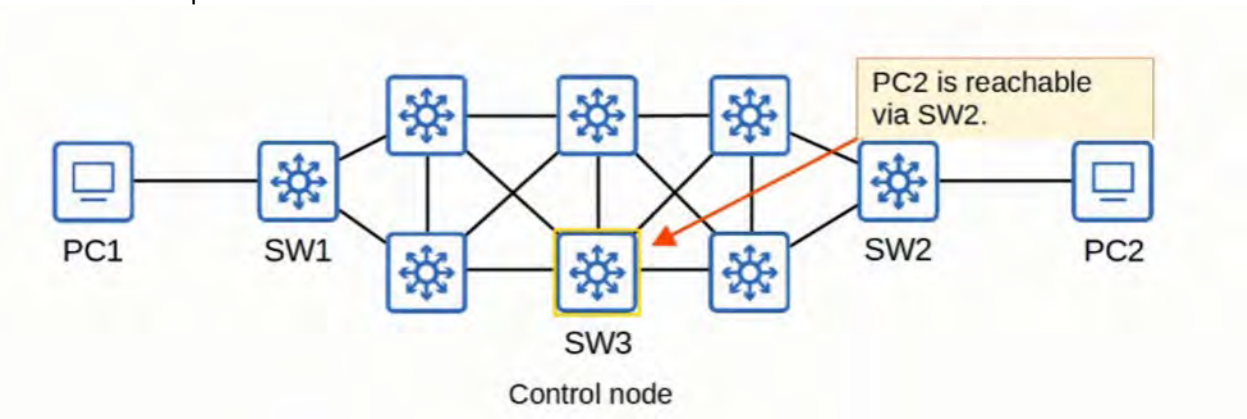
    - BORDER NODES : Connect to devices outside of the SD-ACCESS Domain ; ie: WAN routers
    - CONTROL NODES : Uses LISP (Locator ID Separation Protocol) to perform various control plane functions
- You can add SD-ACCESS on top of the existing network (*brownfield deployment*) if your network hardware and software supports it
    - Google 'Cisco SD-ACCESS compatibility matrix' if you are curious
    - In this case DNA CENTER won't configure the UNDERLAY
- A NEW deployment (*greenfield deployment)* will be configured by DNA CENTER to use the optimal SD-ACCESS UNDERLAY:
    - ALL Switches are LAYER 3 and use IS-IS as their ROUTING PROTOCOL
    - All Links between Switches are ROUTED PORTS. This means STP is not needed
    - EDGE NODES (ACCESS SWITCHES) act as the the DEFAULT GATEWAY of END HOSTS *(Routed Access Layer)*

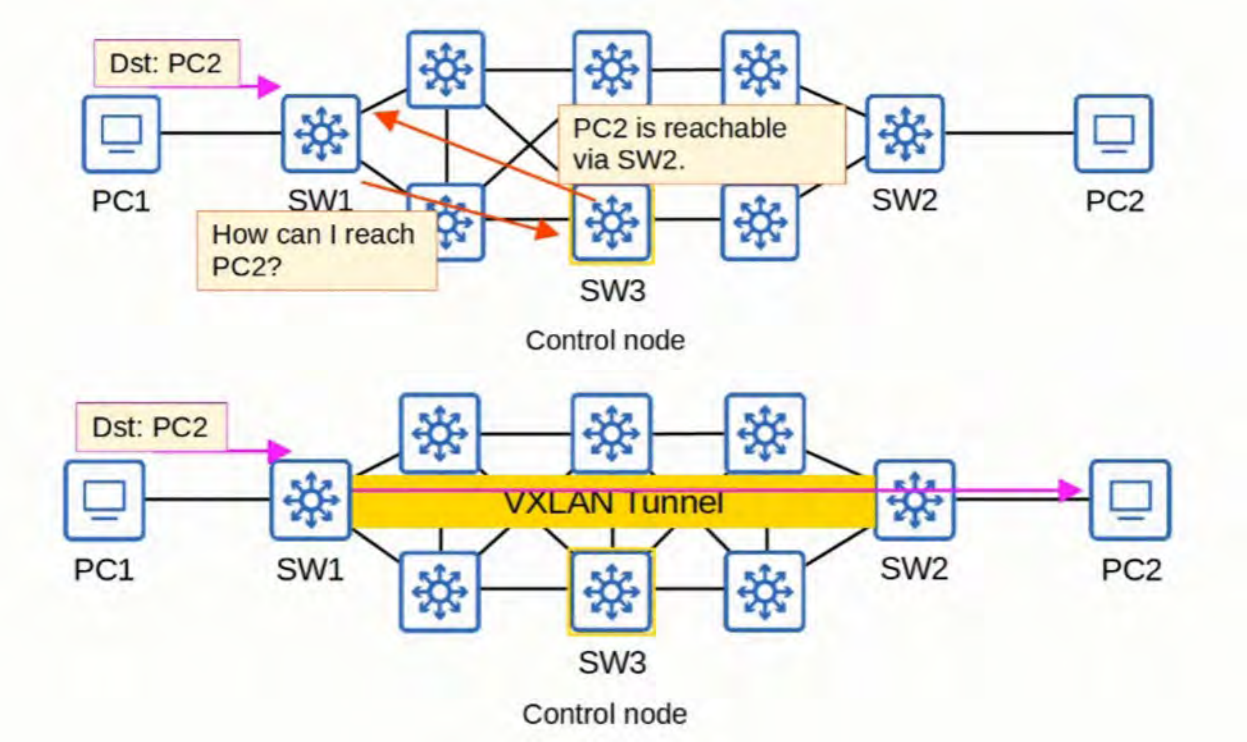




SD-ACCESS OVERLAY

- LISP (Locator ID Separation Protocol) provides the control plane of SD-ACCESS
  - A list of mappings of EIDs (endpoint identifiers) to RLOCs (routing locators) is kept
  - EIDs identify END HOSTS connected to EDGE SWITCHES
  - RLOCS identify the EDGE SWITCH which can be used to reach the END HOST
  - There is a LOT more detail to cover about LISP but I think you can see how it differs from traditional CONTROL PLANE
- Cisco TrustSec (CTS) provides policy control (QoS, Security Policy, etc.)
- VXLAN provides the DATA PLANE of SD-ACCESS





CISCO DNA CENTER

- Cisco DNA Center has TWO MAIN ROLES:
  - The SDN Controller in SD-ACCESS
  - A network manager in a traditional network (non-SD-ACCESS)
- DNA Center is an application installed on Cisco UCS server hardware
- It has a REST API which can be used to interact with DNA Center
- The SBI supports protocols such as NETCONF and RESTCONF (as well as traditional protocols like Telnet, SSH, and SNMP)
- DNA Center enables *Intent-Based Networking* (IBN)
  - The goal is to allow the engineer to communicate their intent for network behavior to DNA Center, and then DNA Center will take care of the details of the actual configurations and policies on devices
- Traditional security policies using ACLs can become VERY cumbersome
  - ACLs can have thousands of entries
  - The intent of entries is forgotten with time and as engineers leave and new engineers take over
- DNA Center allows the engineer to specify the intent of the policy
  - Examples :
    - THIS group of users can’t communicate with THAT group
    - THIS group can access THIS server but not THAT server
  - DNA CENTER will take care of the exact details of implementing this policy

For more details, you can check out sandboxdnac.cisco.com (User: devnetuser, Password: Cisco123!)

DNA CENTER NETWORK MANAGEMENT VS. TRADITIONAL
Traditional Management :

- DEVICES are configured one-by-one via SSH or Console connection
- DEVICES are manually configured via Console connection before being deployed
- Configurations and polices are managed per-device
- New network deployments can take a long time due to the manual labor required
- Errors and failures are more likely due to increased manual effort

DNA CENTER-based Network Management :

- DEVICES are centrally managed and monitored from the DNA CENTER GUI or other applications using it's REST API
- The Administrator communicates their intended network behavior to DNA CENTER, which changes those intentions into configurations on the managed network devices
- Configurations and policies are centrally managed
- Software versions are also centrally managed. DNA CENTER can monitor cloud servers for new versions and then update the managed devices

- New network deployments are much quicker. New devices can automatically receive their configurations from DNA CENTER without manual configuration

Cisco DNA Center provides advantages such as PnP (plug and play), which allows a device to automatically receive its initial configuration from DNAC without requiring manual configuration.

DNAC also supports SWIM (Software Image Management), which ensures that devices remain compliant with software version standards.

63. ANSIBLE, PUPPET, AND CHEF

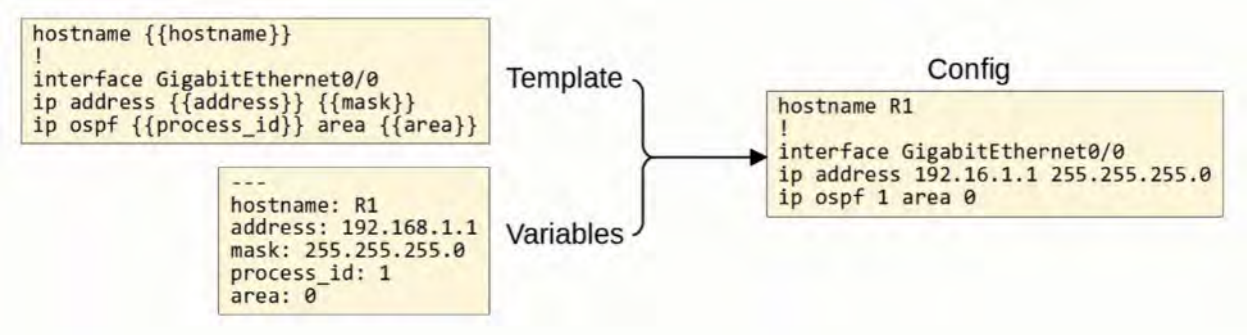
CONFIGURATION DRIFT

- CONFIGURATION DRIFT is when individual changes made over time causes a device's configuration to deviate from the standard / correct configurations as defined by the company
    - Although each device will have unique parts of its configurations (IP Addresses, hostname, etc) most of a device's configuration is usually defined in standard templates designed by the network architects / engineers of the company
    - As individual engineers make changes to devices (for example, to troubleshoot and fix network issues, test configurations, etc), the configuration of a device can drift away from the standard.
    - Records of these individual changes and their reasons aren't kept
    - This can lead to future issues
- Even without automation tools, it is best to have standard configuration management practices.
    - When a change is made, save the config as a text file and place it in a shared folder
        - A standard naming system like (*hostname_yyyymmdd)* might be used.
        - There are flaws to this system, as an engineer might forget to place the new config in the folder after making changes. Which one should be considered the "CORRECT" config?
        - Even if configurations are properly saved like this, it doesn't guarantee that the configurations actually match the standard

---

CONFIGURATION PROVISIONING

- CONFIGURATION PROVISIONING refers to how configuration changes are applied to devices
    - This includes configuring new devices, too
- Traditionally, configuration provisioning is done by connecting to devices one-by-one via SSH
    - This is not practical in large networks
- Configuration management tools like Ansible, Puppet, and Chef allow us to make changes to devices on a mass scale with a fraction of time and effort.
- TWO ESSENTIAL COMPONENTS:
    - Templates
    - Variables



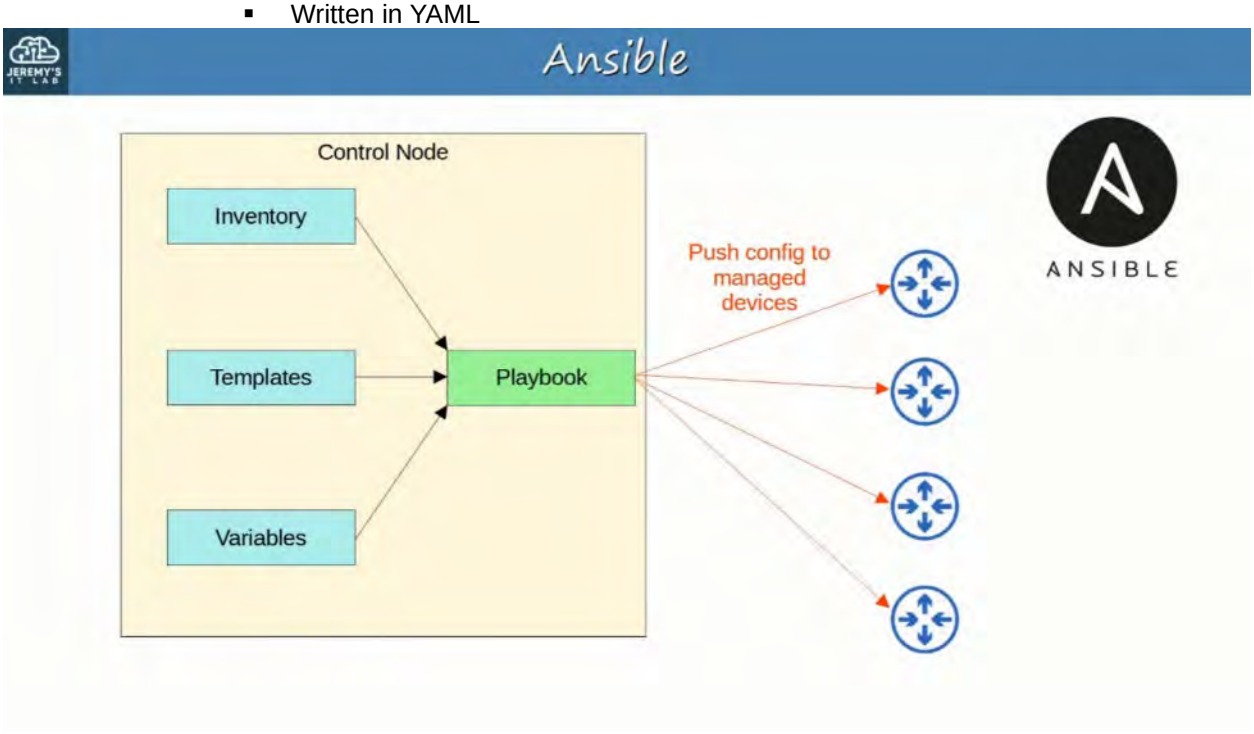
INTRO TO CONFIGURATION MANAGEMENT TOOLS

- CONFIGURATION MANAGEMENT TOOLS are network automation tools that facilitate the centralized control of large numbers of network devices
- The option you need to be aware of for the CCNA are Ansible, Puppet, and Chef
- These tools were originally developed after the rise of VMs, to enable server system admins to automate the process of creating, configuring, and removing VMs
    - However, they are also widely used to manage network devices
- These tools can be used to perform tasks such as :
    - Generate configurations for new devices on a large scale
    - Perform configuration changes on devices (all devices in your network, or certain subset of devices)
    - Check device configurations for compliance with defined standards
    - Compare configurations between devices, and between different versions of configurations on the same device

ANSIBLE

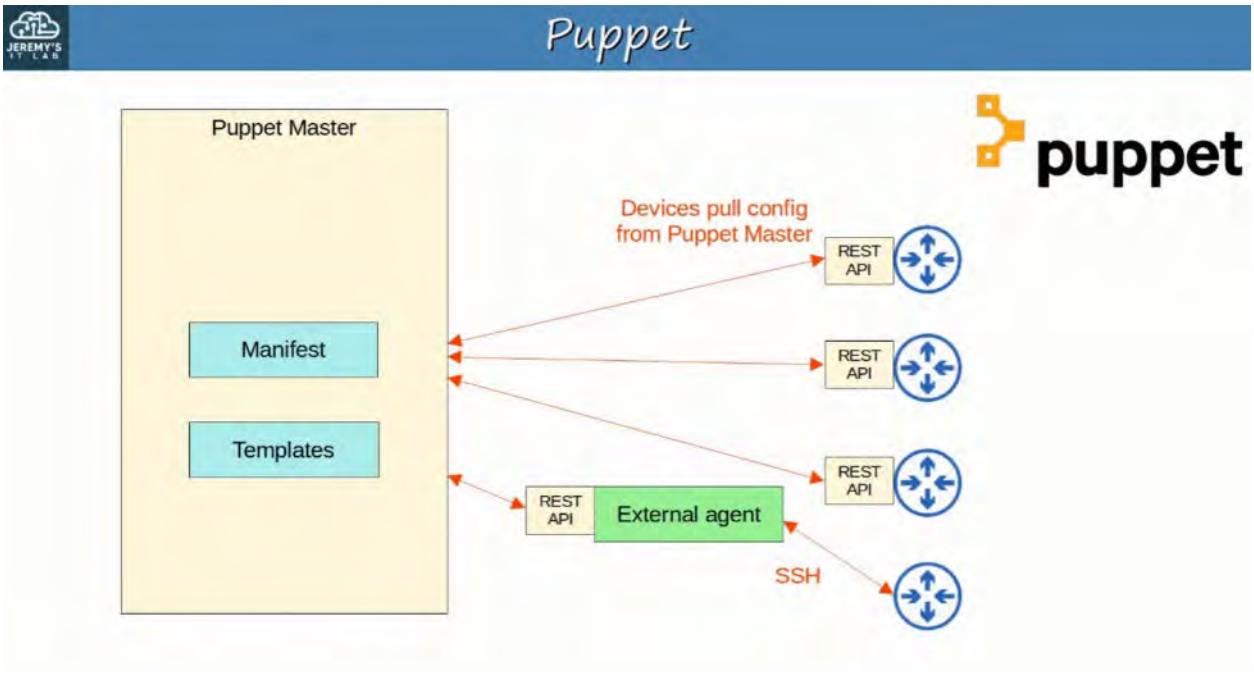
- ANSIBLE is a configuration management tool owned by Red Hat
- Ansible itself is written in Python
- Ansible is *agentless*
    - It doesn't require any special software to run on the managed devices
- Ansible uses SSH to connect to devices, make configuration changes, extract info, etc
- Ansible uses a *push* model. The Ansible server (Control node) uses SSH to connect to managed devices and *push* configuration changes to them
    - Puppet and Chef use a *pull* model
- After installing Ansible itself, you must create several text files:
    - PLAYBOOKS :
        - These files are "blueprints of automation tasks"
        - They outline the logic and actions of the tasks that Ansible should do
        - Written in YAML
    - INVENTORY :
        - These files list the devices that will be managed by Ansible, as well as characteristics of each device such as their device role (Access Switch, Core Switch, WAN Router, Firewall, etc.)
        - Written in INI, YAML, or other formats
    - TEMPLATES :

        - These files represent a device's configuration file, but specific values for variables are not provided.
        - Written in JINJA2 format
    - VARIABLES :
        - These files list variables and their values.
        - These values are substituted into the templates to create complete configuration files.
        - Written in YAML



PUPPET

- PUPPET is a configuration management tool written in RUBY
- Puppet is typically agent-based
    - Specific software must be installed on the managed devices
    - Not all Cisco devices support a Puppet agent
- It CAN be run *agentless,* in which a proxy agent runs on an external host, and a proxy agent uses SSH to connect to the managed devices and communicate with them
- The Puppet server is called the "Puppet master"
- Puppet uses a PULL model (clients "pull" configurations from the Puppet master)
    - Clients use TCP 8140 to communicate with the Puppet master
- Instead of YAML, it uses a proprietary language for files
- Text files required on the Puppet master include:
    - MANIFEST :
        - The file defines the desired configuration state of a network device
    - TEMPLATES :
        - Similar to Ansible templates.
        - Used to generate MANIFESTS



CHEF

- CHEF is a configuration management tool written in RUBY
- CHEF is Agent-Based
  - Specific software must be installed on the managed devices
  - Not all Cisco devices support a CHEF agent
- CHEF uses a PULL model
- The server uses TCP 10002 to send configurations to clients
- Files use a DSL (Domain-Specific Language) based on Ruby
- Text files used by CHEF include:
  - RESOURCES :
    - The “ingredients” in a RECIPE.
    - Configuration objects managed by CHEF
  - RECIPES :
    - The “recipes” in a COOKBOOK.
    - Outlines the logic and actions of the tasks performed on the resources
  - COOKBOOKS :
    - A set of related RECIPES grouped together
  - RUN-LIST :
    - An ordered list of RECIPES that are run to bring a device to the desired configuration state

Chef

https://docs.chef.io/chef_overview/

MEMORIZE THIS CHART FOR THE CCNA

Ansible, Puppet, Chef comparison

	Ansible	Puppet	Chef
Key Files defining actions	Playbook	Manifest	Recipe, Run-list
Communication Protocol	SSH	HTTPS (via REST API)	HTTPS (via REST API)
Key Port	22 (SSH port)	8140	10002
Agent-based/ Agentless	Agentless	Agent-based (or Agentless)	Agent-based
Push/Pull	Push	Pull	Pull