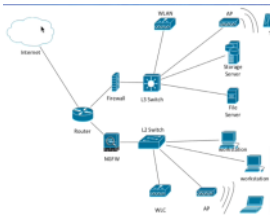
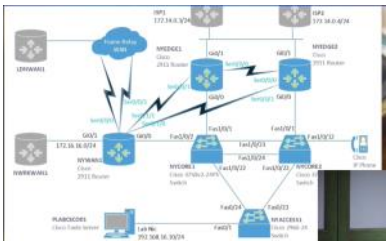
CONCEPTS PART1

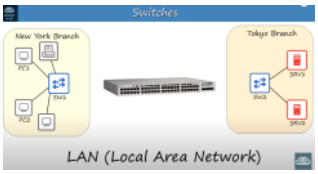
Tuesday, January 17, 2023 6:16 PM



Routers & Switches:



NGFW(next gen firewall) , NGIPS(intrusion prevention systems)



Aggregating clients(hosts) inside of a local area network for switches(can send data from pc1 to pc2 through the switch(eg:catalyst 9200)

Switched cannot connect to other lan in this case from sw1 to sw2 for that we need a router to connect them to internet(sw1->r1,sw2->r2,r1->inteneet->r2)

Routers provide connectivity between lans through internet

Firewalls can be placed outside of the router or inside the router with security rules with which traffic to allow or which not to allow

The network firewall are hardware devices that filter traffic
But host based firewalls are software apps that filter traffic entering machines like a PC

Interfaces and cables:

RJ-45(ethernet cable) - Ethernet which is a collection of network protocols and standards

Bits and bytes (the speed is measure as bits per second)(in harddrive bytes)

Ethernet Standards (copper)

Speed	Common Name	IEEE Standard	Informal Name	Maximum Length
10 Mbps	Ethernet	802.3i	10BASE-T	100 m
100 Mbps	Fast Ethernet	802.3u	100BASE-T	100 m
1 Gbps	Gigabit Ethernet	802.3ab	1000BASE-T	100 m
10 Gbps	10 Gig Ethernet	802.3an	10GBASE-T	100 m

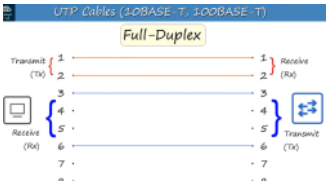
10BASE-T(BASE=baseband signaling,T=twisted pair)

UTP(unshielded twister pair) cables:

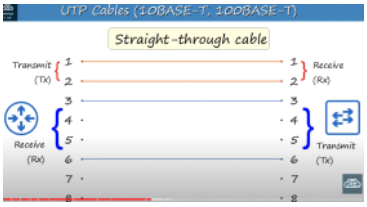
- Twister pair protects against EMI(electromagnetic interference)
- In rj45 in the front it will have 8 pins that correspond to the no of wires in the UTP cables(in pairs)

10BASE-T ,100BASE-T => 2 pairs(4 wires)
1000BASE-T , 10GBASE-T => 4 pairs(8 wires)

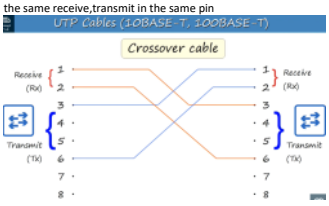
Full deplex communication between pc and switch with 8 wires(in pairs twisted)



Straight through cable for router to switch communication(straight through cables connect from one pin to other pin like 1-1 ,2-2 etc..)



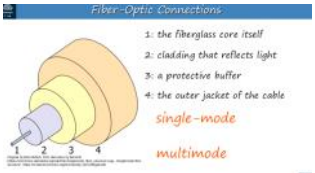
In crossover cable it can connect to pin 1-3,2-6 if the case is between switch and switch.router-router where both ends have the same receive,transmit in the same pin

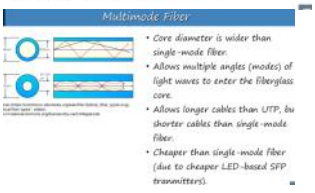


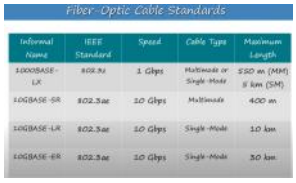
Device Type	Transmit (Tx) Pins	Receive (Rx) Pins
Router	1 and 2	3 and 6
Firewall	1 and 2	3 and 6
PC	1 and 2	3 and 6
Switch	3 and 6	1 and 2

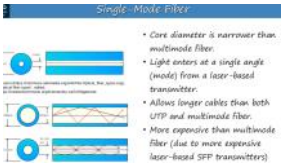
Newer networking cables have the feature called auto mdi-x which detects automatically detect and change the corresponding pins for transmitting and receiving

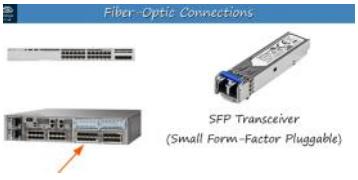
In 1000base-t and 10gbase-t the each pair are bidirectional



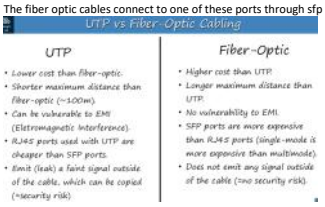








The fiber optic cables connect to one of these ports through sfp transceiver



OSI model:

- HTTP,HTTPS in web browsers are layer 7 application layer
- Translation between application and network formats happen at layer 6 presentation layer
  And it does also the encryption and decryption
- Maintaining sessions for the communications isis the job of layer 5 session

Application developers work with these top 3 layers and connect to network and pass it to network engineers

- The layer 4 transport layer segments data into larger pieces of data into smaller segments and reasssembles
  Provides host-host,end-end,process-process communication
- The layer 3 network layer provides connectivity between end hosts on different networks and provides logical addressing , path selection between src,dest(routers)

Data+l4 header => segment
Data+l4 header+l3 header = > packet
L2 trailer +Data+l4 header+l3 header + l2 header = Frame

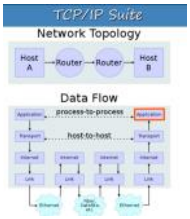
- The layer 2 layer data link defines how data is formatted for transmission over physica medium
  And also uses some addressing like network layer (swithc is at layer2)
- In Layer 1 physical the digitsl bits are converted into electrical(wired) or radio(wireless)
  Defines voltage levels,maximum transmission distances,cable specifications etc..

Data,segment,packet,frame,bits are protocol data units(PDU's)

TCP/IP(currently used one)

- L7,6,5 => L4 (Application layer)
- L4 => L3(transport layer)
- L3 => L2(internet layer)

- L2,1=> L1(Link layer)



IOS CLI cisco:
(os in cisco)
Also it has GUI for cisco devices

For connecting to the cisco device form laptop we need to connect
to the console port(RJ45,USB Mini-B) in that device(also remote access is available)

Rollover cable is used for console port connection to laptop
Pins are connected like 1-8,2-7,3-6,4-5,5-4,6-3,7-2,8-1

Use putty to connect to cli use the connection type as Serial(there are raw telnet rlogin ssh serial as
Connection types)

In the serial connection type setting there are data bits,stop bits .
If data bits is 8 , stop bits is 1 there is 1 stop bit sent for every 8 data bits
Also speed(baud rate) can be configured,parity option to control errors,
Flow control to control the flow from transmitter to receiver

Inside the cli Router> (> means user exec mode,Router -> hostname)
Router> enable
Router# (# is priviledged exec mode)
This givrs complete access to view device configuration,restart the device etc..
We cant change the configurations in each mdoe but can change the time and save config etc..

? -> command to list the commands

e? -> possible commands starts with e

configure terminal -> to enter global config mode(needs priveledged mode)
Can also type conf t since it is only the command with starting with conf , t for terminal

enable password ? (gives the possible word after that command(if the output is <cr> there is no further options available))
enable password CCNA -> sets the password CCNA
(when entering enable command next time it will ask password)
exit -> logout of the modes

- There are two separate configuration files kept on the device at once

1) Running-config = current active config file on which you edit as you enter cmds in cli
2) Startup-config = config file that loaded upon restart of the device

show running-config -> see details in that file
show startup-config -> which will not be available

write,write memory -> wrtie the config
copy running-config startup-config -> copy to startupconfig file

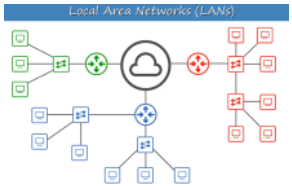
service password-encryption -> encrypts the password in config file(requires global config mode)

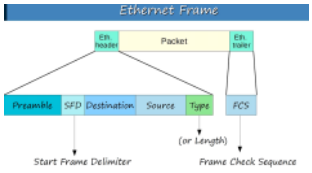
In the config file it will display as enable password 7 03284f23498 (here 7 means the encryption protocol cisco used to
encrypt

enable secret Cisco -> secure way of encrypting password
do sh run(show runtime-config from global config mode)
(
enable secret 5(MD5) $8734724287496828428749234
enable password 7 873689746) <- this has not effect if we use secret

no service password-encryption <- cancelling commands(disable)

Hostname T1 -> set hostname
Enable secret >>> enable password





(usually preamble,sfd not considered part of the ethernet header)
Preamble
- length: 7 bytes(56 bits)
- Alternating 1's and 0's
- 10101010 * 7
- Allows devices to synchronize their receiver clocks

SFD
- Start frame delimiter
- Length: 1 byte
- 10101011
- Marks the end of the preamble and the beginning of the rest of frame

Destination,source -> sending and receiving the frame and consists of MAC addresses
MAC
- 6 byte (48bit(address of the physical device

Type/Length is 2 byte(16 bit) , value of 1500 or less indicates length of the encpsulated packet(in bytes)
Value of 1536 or greater indicates the type(ipv4 or ipv6)

FCS:
- Frame check sequence
- 4 bytes(32 bits)
- Detects corrupted data by running a CRC algo over the received data(cyclic redundancy check)

Header+trailer => 26 bytes

MAC ADDRESSES:
- 6 byte (48bit(address of the physical device))
- A.K.A 'burned in address'
- Globally unique
- The first 3 bytes are the OUI(organizationally unique identifier) which is assigned to the company making the device
- The next 3 bytes are unique to the device itself

Eg: AAAA.AA00.0001 => AAAA.AA(OUI) , 00.0001(UNIQUE TO DEVICE)

Decimal:

MAC ADDRESSES:

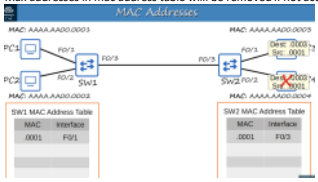
- 6 byte (48bit(address of the physical device))
- A.K.A 'burned in address'
- Globally unique
- The first 3 bytes are the OUI(organizationally unique identifier) which is assigned to the company making the device
- The next 3 bytes are unique to the device itself

Eg: AAAA.AA00.0001 => AAAA.AA(OUI) , 00.0001(UNIQUE TO DEVICE)

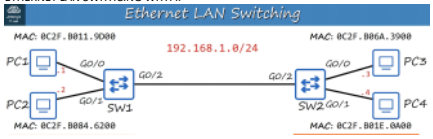
- Written as 12 hexadecimal characters

Frames destined to a single target is called unicast frame

When pc1 sends frame to pc2 through switch then the switch dynamically learns the transfer I puts a record into its mac address table with the interface with which it received the frame and the corresponding mac.
Since it is the first time the swithc receives the packet through its interface the mac table has that entry but it does not know the mac address for pc and through which interface since it is not in the max address table of it so this scenario is called unknown unicast frame . So FLOOD is done to send it to all the interfaces and the pc's that are are meant to be sent will drop the frame and the pc whicih is meant will accept.
When the pc's send the packet to pc2 then one by one it will be added to mac address table then it becomes known unicast frame => FORWARD is done
Max addresses in mac address table willl be removed if not active for 5 minutes



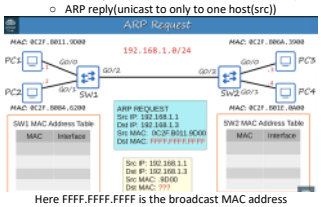
ETHERNET LAN SWITHCING WITH IP



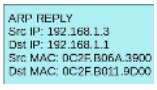
Switch is at layer 2 so it needs only mac instead of IP so we need to use ARP to define the target MAC

ARP

- Address resolution protocol
- Used to discover the layer 2 address(MAC) of a known layer 3 address(IP)
- Consists of 2 messages
  - ARP request(broadcast to all hosts in network)
  - ARP reply(unicast to only to one host(src))

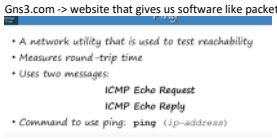


Here FFFF.FFFF.FFFF is the broadcast MAC address



In our pc if we give in cmd (arp -a (to view the arp table))
In the output if its type is statis then it is default entries but if it is dynamic then it is learned via ARP

Gns3.com -> website that gives us software like packet tracer that has actual CISCO IOS installed virtually BUT WITH PAID



The first ping would fail to do the arp request and then it will succeed

Show arp -> arp table in cisco cli (priveledged mode)

Use wireshark to see the traffic between networks

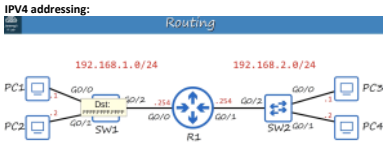
Show mac address-table -> see mac address table in cisco cli(priveledged mode)
Clear mac address-table dynamic -> to clear dynamic mac address entries
Clear mac address-table dynamic address addressmac
Clear mac address-table dynamic interface Gi0/0

0x0800 -> 0x to indicate hexadecimal

Min size for an ethernet frame is 64 bytes , minimum payload size is 46 bytes, if less than this padding bytes are added

VLAN,MAC ADDRESS,TYPE,PORTS are the fields in the show mac address-table

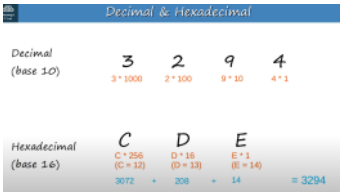
IPV4 addressing:

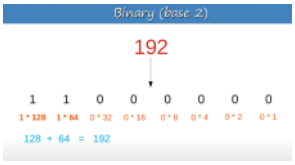


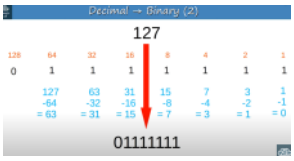
Here the since the router is in place these are seperated into two lans
And the ip for router is taken from each lan to put it to the router's interface
Here the broadcast cant happen between router and switch

Decimal:
0 10
1 11
2 12
3 13
4 etc..
5
..9

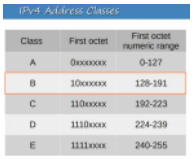
Hexadecimal: (just continues to increment the tens,hundreds and etc.. And gives the hexadecimal values preceding it)
0 10
1 11
2 12
3 13
4 14
..9 25
A 1A
B 1B
..F 1F







192.168.1.254/24 -> here /24 means the first 24 bits represent the network portion of the address and remaning 8 bits represent end hosts(32 bits(8bits*4))



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

D is reserved for multicast addresses,E is reserved for experimental uses
In class A the 0-127 means only 1-126 since the 127 is taken for loopback addresses

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

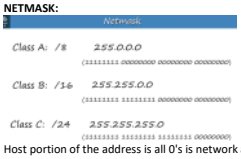
In class A 0 is also reserved
Loopback addresses:
- Address range 127.0.0.0 - 127.255.255.255
- Used to test the network stack(think OSI,TCP/IP model) on the local device
- It is like localhosts ranges
- If we ping to one of its address then the max,min,avg waiting time will be 0ms since it sends/receives to itself

To find the number of ip addresses for networks possible we can do 2^prefixlength-1(for A),s* length-2(for B),2 ^length03(for C)

For hosts 2^(totalbits-prefixlength)

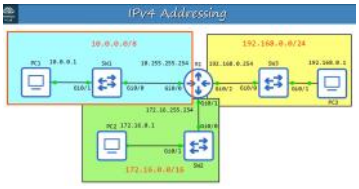
In the host/network count the last two hosts will be for broadcast addresses

NETMASK:



Host portion of the address is all 0's is network address,
Host portion of the addres sis all 1's = broadcast address
Both cant be assigned to a host

Example:
192.168.1.0/24
Host portion is last 8 bits that meas 0-255 in last octet=256 hosts
Maximum hostsper network = 2^8(256)-2(network,broadcast address)=254



In the cli of router
- en
- show ip interface brief

In the above command output in the status if we get administratively down then interace has been disabled with the shutdown command (default for cisco router interfaces)
(not default for cisco switch)
Status column refers to layer 1 status
Protocol column refers to layer 2 status

#conf t
(config)#interface gigabitinternet 0/0(or interface gigabitethernet0/0)(interface config mode)
Or (in g0/0)
(config-if)#ip address 10.255.255.254 ?
(config-if)#ip address 10.255.255.254 255.0.0.0
(config-if)#no shutdown
(interafce should change state to up)

R1# show interfaces g0/0
R1# show interfaces description
R1(config)#int g0/0
R1(config-if)#description ## to SW1 ##
R1(config)#int g0/1
R1(config-if)#description ## to SW2 ##
Need to copy runtime config to startup config to save it to memory and at next startup

CONCEPTS PART2

Monday, February 6, 2023 4:22 PM

Switch interfaces:

- The switch status , protocol in show ip interface brief command since once it gets connected to a device it will be up since it does not have shutdown command applied by default

SW1# show interfaces status
SW1# conf t
SW1(config)#int f0/1
SW1(config-if)#speed 10/100/auto (to force 100mbps operation)
SW1(config-if)#duplex full/half/auto (to force duplex operation)

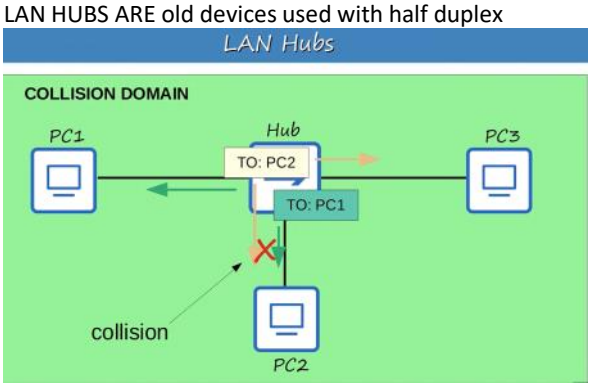
(if duplex or speed is displayed as a-100,a-full(auto negiotiated when transferring)

SW1(config)#interface range f0/5 - 12 (5,6,7,8,9,10,11,12)
Also can be written as (int range f0/5 -6 , f0/9 - 12)
SW1(config-if-range)#shutdown

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

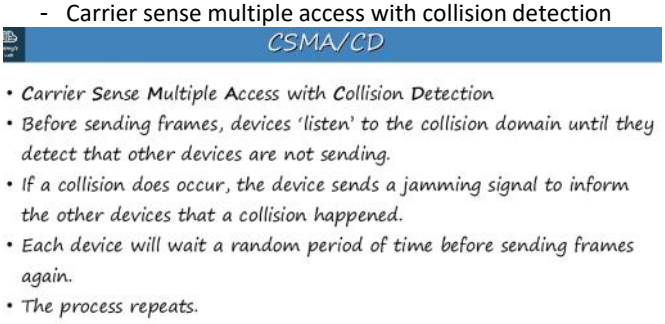
- **Full duplex**: The device **can** send and receive data at the same time. It does not have to wait.

LAN HUBS ARE old devices used with half duplex

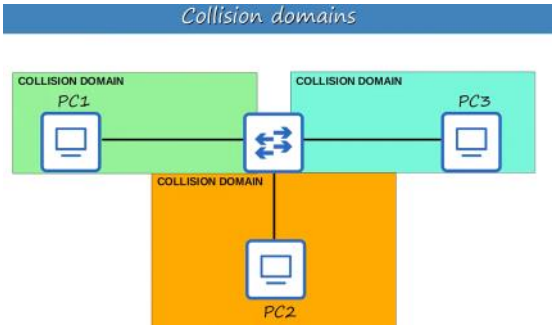


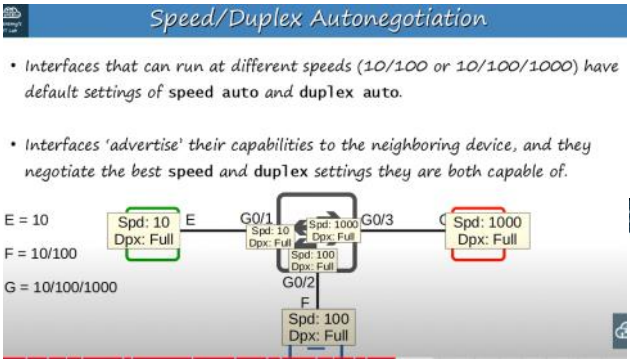
CSMA/CD:

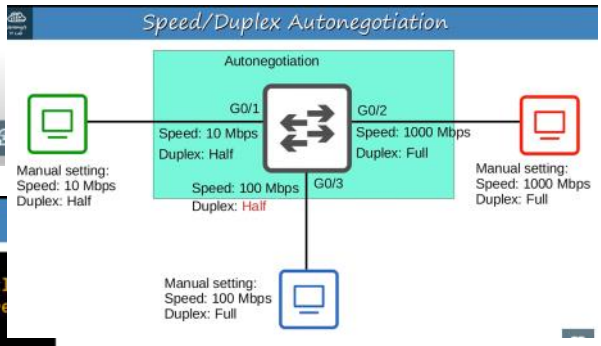
- Carrier sense multiple access with collision detection

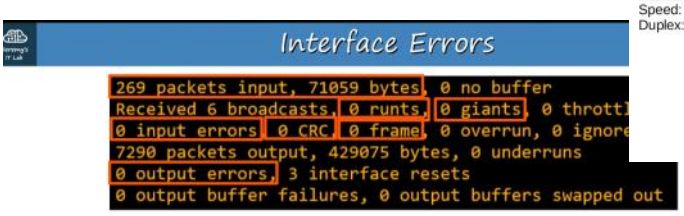


THE BELOW IS FOR SWITCH







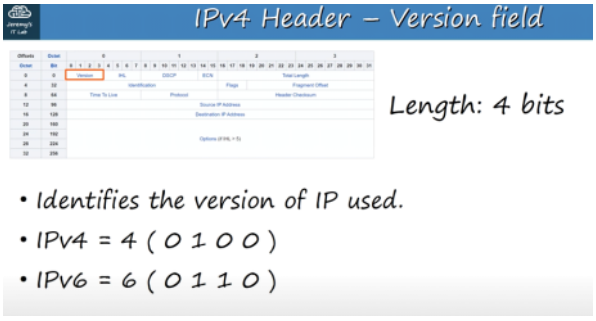


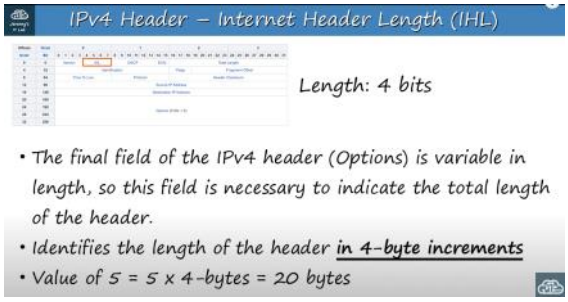
- **Runts**: Frames that are smaller than the minimum frame size (64 bytes)
- **Giants**: Frames that are larger than the maximum frame size (1518 bytes)
- **CRC**: Frames that failed the CRC check (in the Ethernet FCS trailer)
- **Frame**: Frames that have an incorrect format (due to an error)
- **Input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

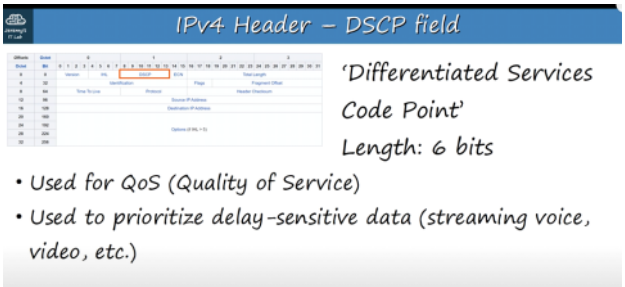
Show interfaces (checking the interface errors etc..)

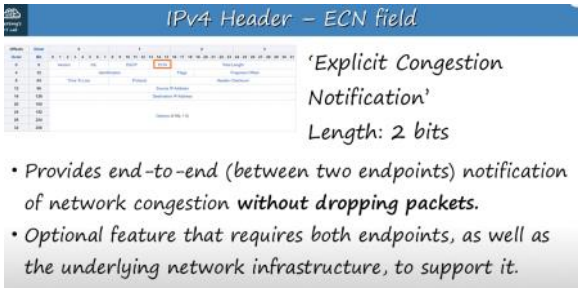
In the ipv4 header the identification field we can have MTU(maximum transmission units) as 1500 bytes which is the max size of an ethernet frame
recommended TTL in ipv4 header is 64

IMP IPV4 HEADER FIELDS EXPLAIN









IPv4 Header – Total Length field

Length: 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 (=IPv4 header with no encapsulated data)
- Maximum value of 65,535 (maximum 16-bit value)

IPv4 Header – Identification field

Length: 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)

IPv4 Header – Flags field

Length: 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

Subtitles/closed captions (c)

IPv4 Header – Time To Live field

Length: 8 bits

Recommended default TTL is 64.

- A router will drop a packet with a TTL of 0
- Used to prevent infinite loops
- Originally designed to indicate the packet's 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.

IPv4 Header – Fragment Offset field

Length: 13 bits

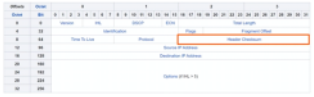
- Used to indicate the position of the fragment within the original, unfragmented IP packet.
- Allows fragmented packets to be reassembled even if the fragments arrive out of order.

IPv4 Header – Protocol field

Length: 8 bits

- Indicates the protocol of the encapsulated L4PDU
- Value of 6: TCP
- Value of 17: UDP
- Value of 1: ICMP
- Value of 89: OSPF (dynamic routing protocol)

IPv4 Header – Header Checksum field



Length: 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 the header.
- If they do not match, the router drops the packet.

Miniplayer (i)

IPv4 Header – Options fields

Length: 0 - 320 bits

- Rarely used.
- If the IHL field is greater than 5, it means that Options are present.

Field	Size (bits)	Description
Copied	1	Set to 1 if the options need to be copied into all fragments of a fragmented packet.
Option Class	2	A general options category. 0 is for "control" options, and 2 is for "debugging and measurement". 1 and 3 are reserved.
Option Number	5	Specifies an option.
Option Length	8	Indicates the size of the entire option (including this field). This field may not exist for simple options.
Option Data	Variable	Option-specific data. This field may not exist for simple options.

Ping 192.168.1.2 df-bit (df-bit meansdont fragment the packet into multiple frame/packets)
Ping ip size 10000 df-bit(here it will not be success)

The **Options** field can vary in length from 0 bits to 320 bits. The other fields are fixed-length. Although the **Total Length** and **IHL** fields are used to represent the variable length of the IPv4 header and packet, the fields themselves are fixed in length.

What is routing?

- **Routing** is the process that routers use to determine the path that IP packets should take over a network to reach their destination.
  - → Routers store routes to all of their known destinations in a **routing table**.
  - → When routers receive packets, they look in the **routing table** to find the best route to forward that packet.
- There are two main routing methods (methods that routers use to learn routes):
  Dynamic Routing: Routers use *dynamic routing protocols* (ie. OSPF) to share routing information with each other automatically and build their routing tables.
  - → We will cover this later in the course.

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

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

What is routing?

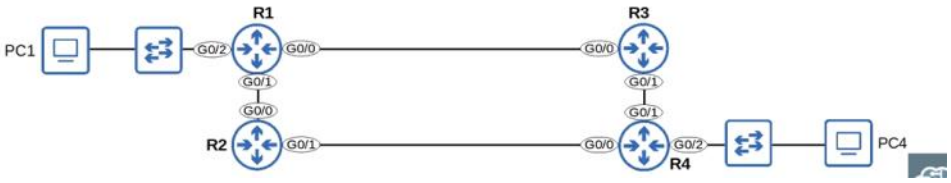
- **Routing** is the process that routers use to determine the path that IP packets should take over a network to reach their destination.
  → Routers store routes to all of their known destinations in a **routing table**.
  → When routers receive packets, they look in the **routing table** to find the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

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

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

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

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



WAN is wide area network is a network that extends over a large geographical area

R1# show ip route (shows the routing table)

Connected and Local routes

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

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

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

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**.

- `192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks`
  → In the routing table, there are two routes to *subnets* that fit within the 192.168.1.0/24 Class C network, with two different netmasks (/24 and /32).

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:

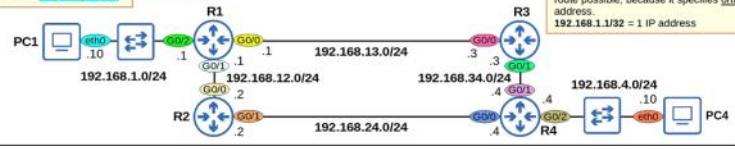
```
iface eth0 inet static
        address 192.168.4.10/24
        gateway 192.168.4.4
```

- 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



Routing Packets: Static Routes

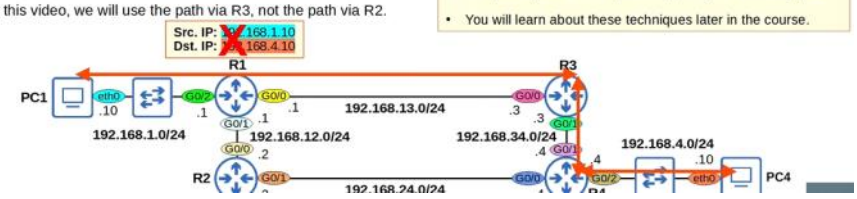
- When R1 receives the frame from PC1, it will de-encapsulate it (remove L2 header/trailer) and look at the inside packet.
- It will check the routing table for the most-specific matching route:

```
     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
```

- 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.

- 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.

Src. IP: [illegible]168.1.10
Dst. IP: [illegible]168.4.10



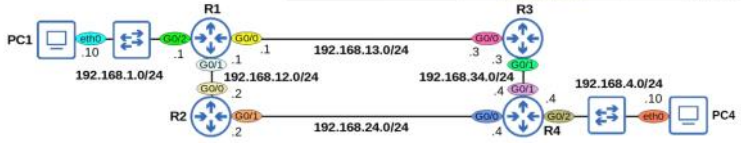
Static Route Configuration

- 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



R1(config)# ip route ip-address netmask next-hop
R1(config)# ip route 192.168.4.0 255.255.255.0 192.168.13.3
R1(config)# do show ip route

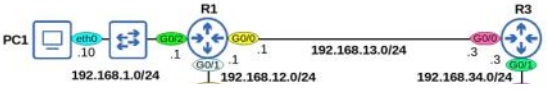
Static Route Configuration (R1)

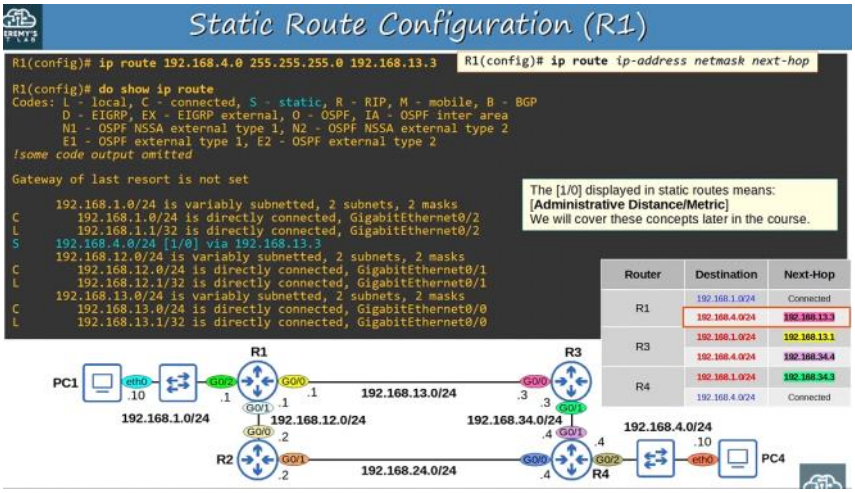
```
R1(config)# ip route 192.168.4.0 255.255.255.0 192.168.13.3

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.

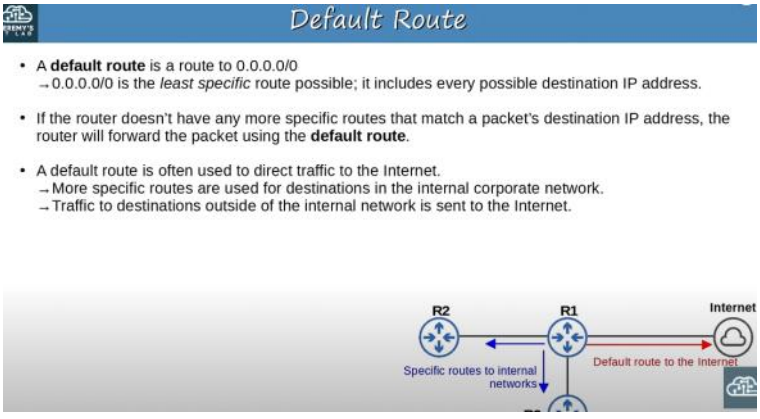
Router	Destination	Next-Hop
R1	192.168.1.0/24	Connected
	192.168.4.0/24	192.168.13.3
R3	192.168.1.0/24	192.168.13.1
	192.168.4.0/24	192.168.34.4
R4	192.168.1.0/24	192.168.34.3
	192.168.4.0/24	Connected

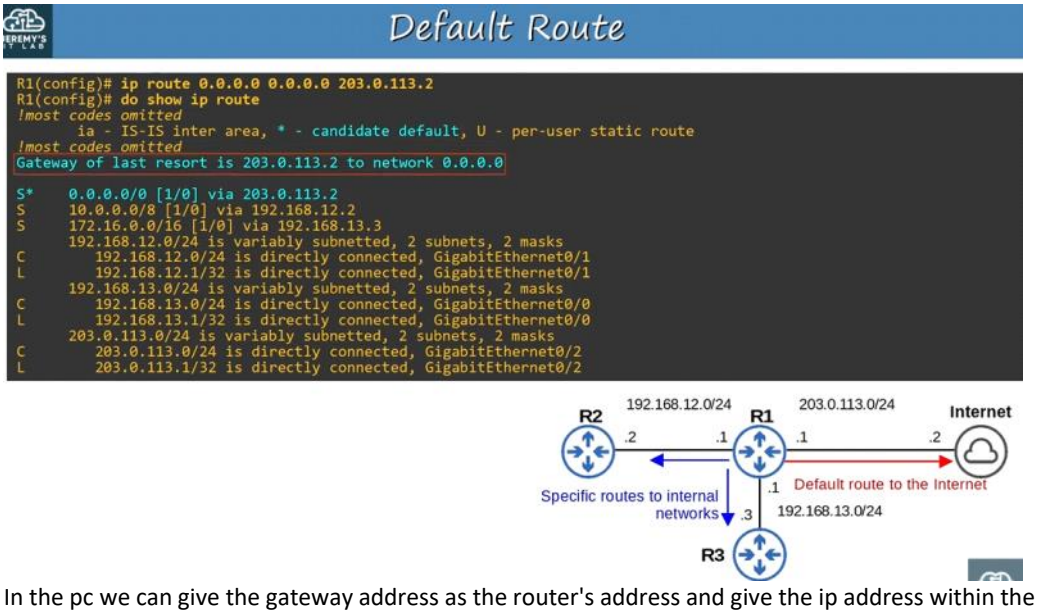




Connecting routes with the exit interface option
R2(config)# ip route 192.168.1.0 255.255.255.0 g0/0
(also)
R2(config)# ip route 192.168.1.0 255.255.255.0 g0/0 nexthopipd

- 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.





In the pc we can give the gateway address as the router's address and give the ip address within the range we define for the lan

In each iteration like wuther from a pc -> router or router to router etc.. In each one arp request will be send to the corresponding device and after getting corresponding mac addresses then it all gets connected

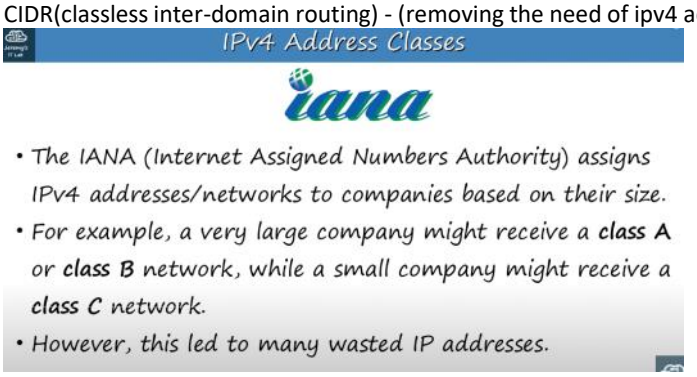
R1# mac-address kjjl.kjhh.9090

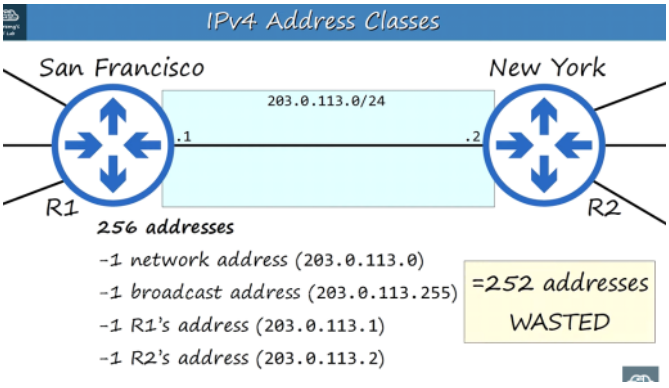
Ipconfig /all (to see all details including mac address)

SUBNETTING:

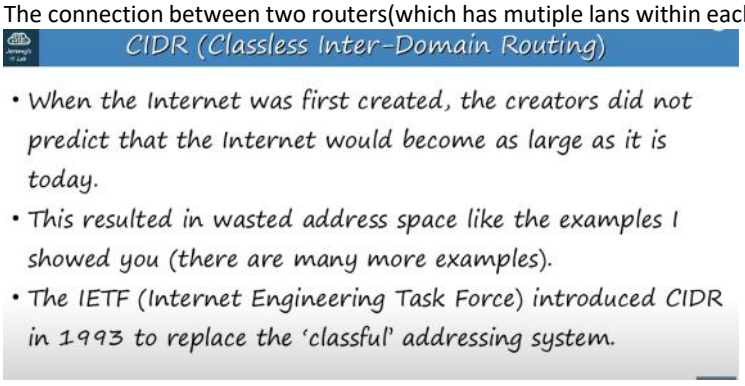
- Dividing large - smaller networks

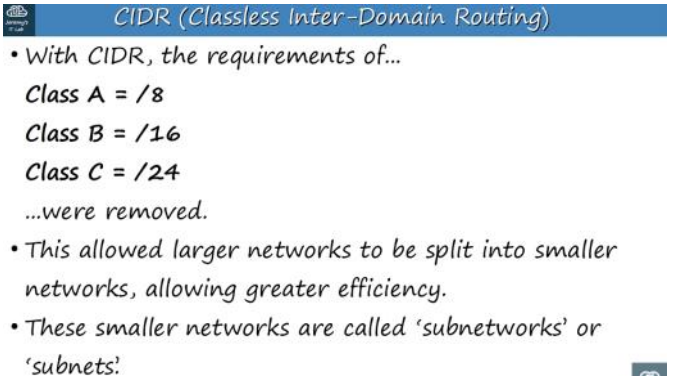
CIDR(classless inter-domain routing) - (removing the need of ipv4 address classes)

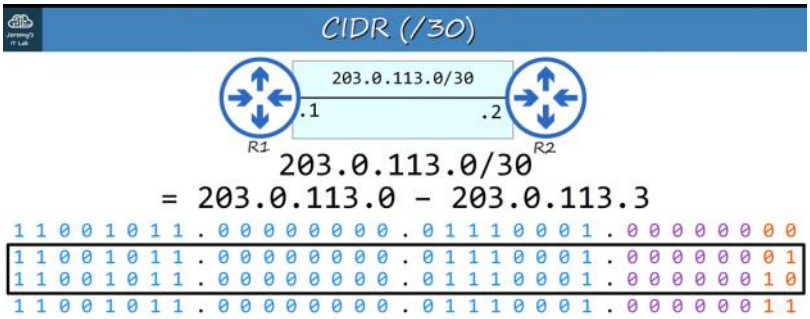




The connection between two routers(which has mutiple lans within each router) is called point -point network







In /31 cidr when allocating for routers in a point to point connection there is don't need to have broadcast address,network address. So we can break it and assign those 2 addresses to both routers with /31 but when we configure it with cisco it will show warning

To find the network address,broadcast address try to put the host portion fully 0's for network,1's for broadcast

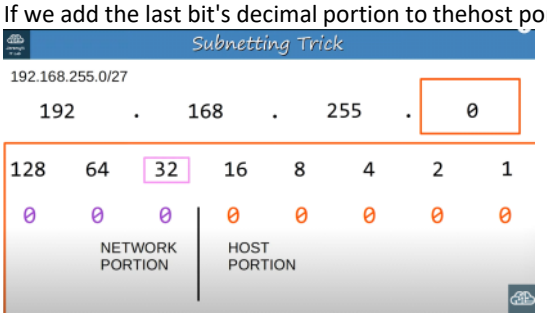
NETWORK ADDRESS (192.168.1.0/24)

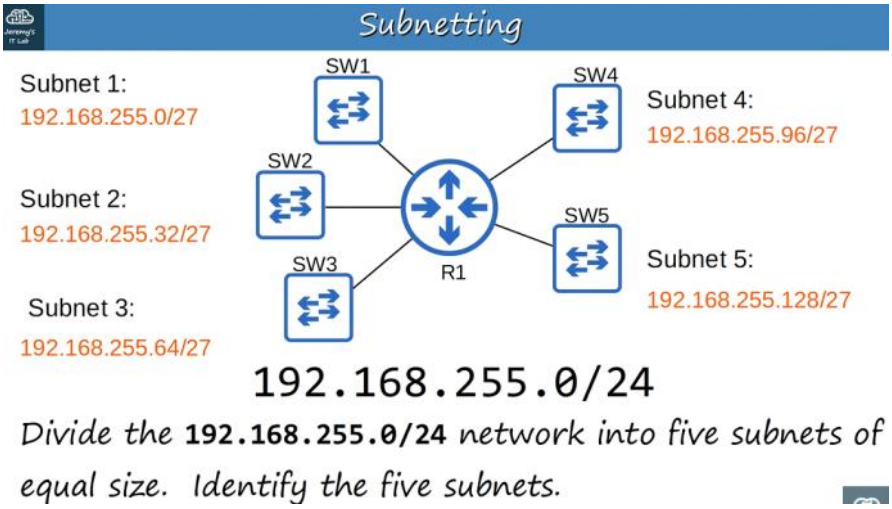
Subnet1:(since we need 45 hosts we use /26)
192.168.1.0/26 => 192.168.1.0(network address)
192.168.1.63(broadcast address)
Subnet2:(the next address after the broadcast addresses is network address of next subnet)
192.168.1.64/26 =>192.168.1.64(network address)
192.168.1.127(broadcast address)

…… and so on

Formula for subnets (2^x(x=number of borrowed bits from network)

If we add the last bit's decimal portion to thehost portion then we will get the next subnet starting address





To identify the subnet from an ip address we can change the host portion to 0's and find the subnet with that /

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)

Subnets/Hosts (Class B)

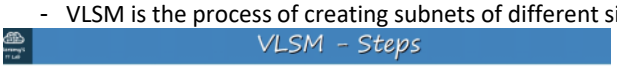
Prefix Length	Number of Subnets	Number of Hosts
/17	2	32766
/18	4	16382
/19	8	8190
/20	16	4094
/21	32	2044
/22	64	1022
/23	128	510
/24	256	254

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)

Borrowing bits means borrowing bits from the given ipv4 cidr

VARIABLE LENGTH SUBNET MASKS (VLSM)

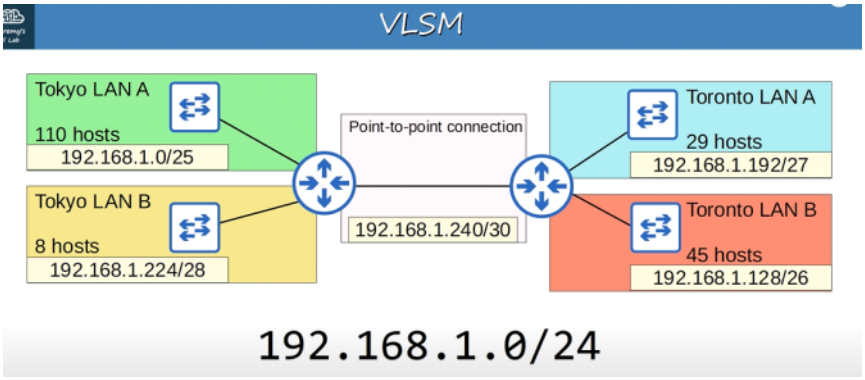
- Till now we have used FLSM(fixed length subnet masks)
- This means that all subnets use the same prefix length
- VLSM is the process of creating subnets of different sizes, to make your use of network addresses more efficient



1) Assign the largest subnet at the start of the address space.

2) Assign the second-largest subnet after it.

3) Repeat the process until all subnets have been assigned.



Subnetting practice:

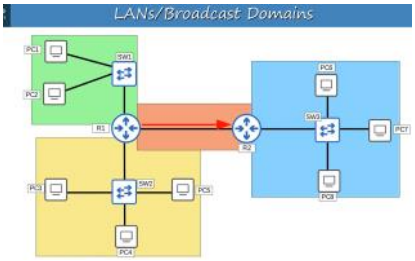
- http://www.subnettingquestions.com
- http://subnetting.org/
- https://subnettingpractice.com/

CONCEPTS PART3

Tuesday, February 7, 2023 6:23 PM

LOCAL AREA NETWORK:

- Group of devices in a single location
- Or
- Single broadcast domain, including all devices in that broadcast domain
- A broadcast domain is the group of devices which will receive a broadcast frame (destination MAC FFFF.FFFF.FFFF) sent by any one of the members)



VLAN:

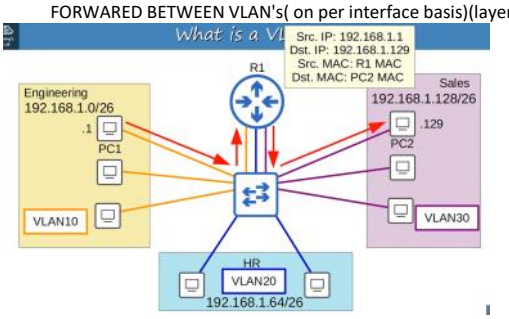
- Lots of unnecassary broadcast traffic can reduce network performance

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

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

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

- In switch we can assign specific hosts as VLAN's SO FOR BROADCAST/UNKNOWN UNICAST TRAFFIC WILL NOT BE FORWARED BETWEEN VLAN's(on per interface basis)(layer2)



(for the above confiuration to be successful we need to make a vlan for the interfaces that connect to pc and also the interface which connects to router as a single vlan and in the pc's give the default gateway as that particular interface to which switch is connected for that vlan)

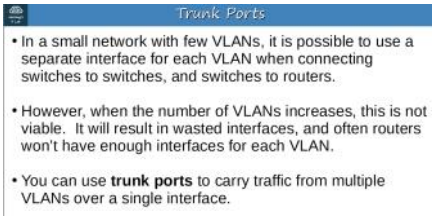
SW1# show vlan brief (**default** vlan configured for all interfaces) (there will be also other vlan's listed but they are old and theese cannot be deleted(usually 4 vlans))

SW1(config)# interface range g1/0 -3
SW1(config-if-range)# switchport mode access
SW1(config-if-range)# switchport access vlan 10

(repeat the above 3 steps for multiple range of interfaces based on our needs)

Access port is a switchport which belongs to a single VLAN , and usually connects to end hosts like PC's
Switchports which carry multiple VLANs are called **trunk ports**

SW1(config)# vlan 10 (to siwitch to the vlan/create a vlan)
SW1(config-vlan)# name ENGINEERING
SW1(config-vlan)#vlan 20
SW1(config-vlan)# name HR (to change the name of the vlan)

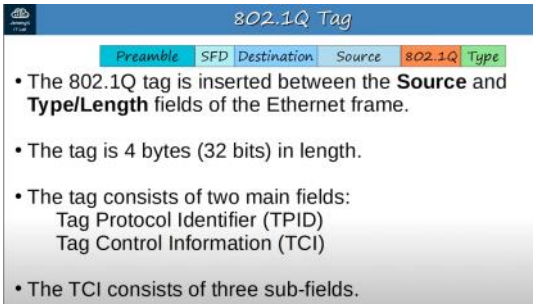


Trunk ports = 'tagged' ports

Access ports = 'untagged' ports

Trunking protocols:

- ISL(Inter-Switch Link)
  - Old cisco propretory protocol created before IEEE 802.1Q
- IEEE 802.1Q is an industry standard protocol created by IEEE(institute of electrical and electronics engineers) (modern,real world application)



802.1Q tag format

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

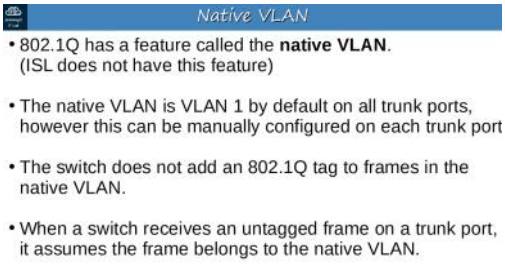
TPID:
- Tag protocol identifier
- 16 BITS IN LENGTH
- Always set to a value of 0x8100. This indicates that frame is 802.1q-tagged

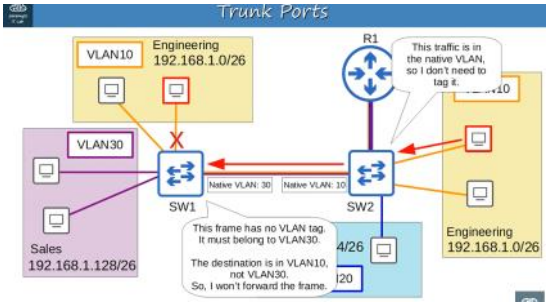
PCP
- Priority code point
- 3 bits in length
- Used for class of service, which priotorizes important traffic in congested networks

DEI
- Drop eligible indicator
- 1 bit in length
- Indicate frames that can be dropped if network is congested

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
- First and last vlans are reserved(0,4095)
- So range is 1-4094
- ISL protocol also uses 1-4094

Normal VLANs : 1-1005
Extended VLANs 1006 - 4094(old devices may /cannot use extended vlan)





If pc from VLAN10 sends to switch having native VLAN10 then it will forward them to it, but if native VLAN does not match the packet it will discard , also if VLAN10 is sending data to VLAN30 with VLAN30 tagged it will discard since it is a native vlan(should be untagged)
Source should be same as target

SW1(config)# interface g0/0
SW1(config-if)# switchport mode trunk
(this will not be allowed if the switch supports both isl,dot1q, and set to auto(negotiate), so need to manually set to dot1q or isl)

SW1(config-if)# switchport trunk encapsulation dot1q
SW1(config-if)# switchport mode trunk
(accepted)

SW1# show interfaces trunk

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

SW1(config-if)# switchport trunk allowed vlan add 20
SW1(config-if)# switchport trunk allowed vlan remove 20
SW1(config-if)# switchport trunk allowed vlan all(default state)

SW1(config-if)# switchport trunk allowed vlan except 1-5,10

SW1(config-if)# switchport trunk allowed vlan none

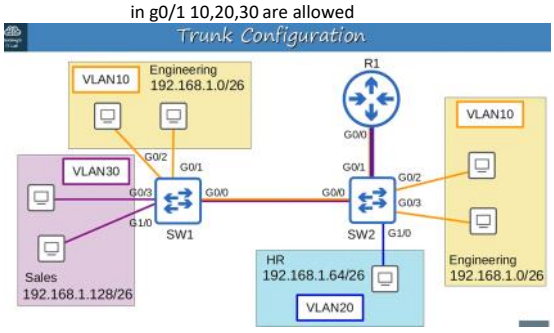
For security purposes, it is best to change the native VLAN to an **unused VLAN**.
(network security will be explained more in-depth later in the course)
Make sure the native VLAN matches on between switches

SW1(config-if)# switchport trunk native vlan 1001

The **show vlan brief** command shows the access ports assigned to each VLAN, NOT the trunk ports that allow each VLAN.

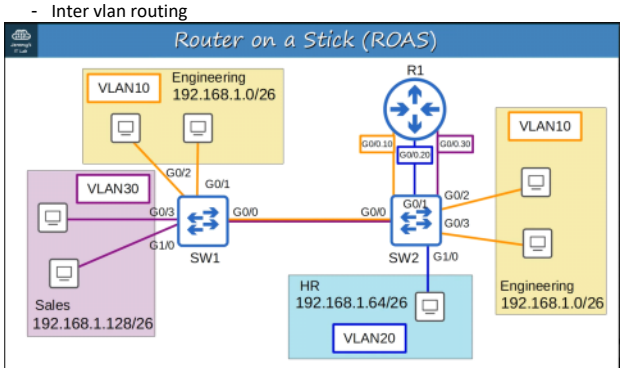
Use the **show interfaces trunk** command instead to confirm trunk ports.

In switch 1 now in g0/0 10,30 are allowed,
In switch 2 now in g0/0 10,30 are allowed
in g0/1 10,20,30 are allowed



ROUTER ON A STICK(ROAS)

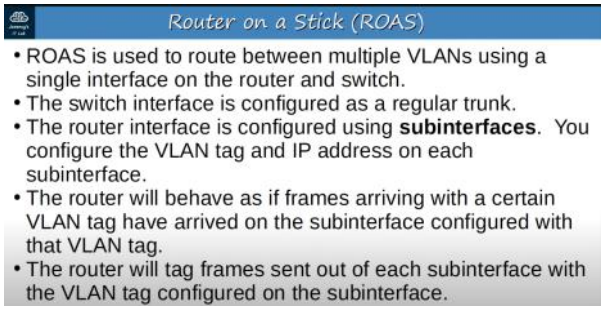
- Inter vlan routing




```
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)#
```

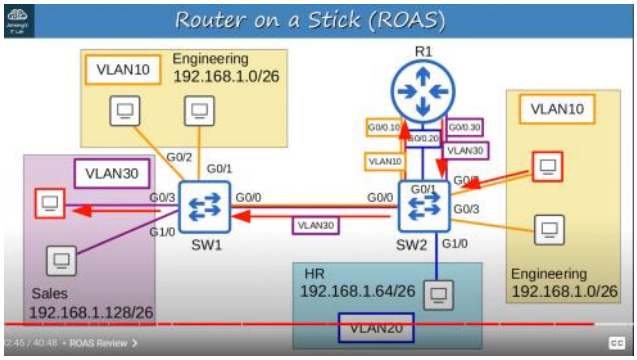
The **encapsulation dot1q 10** will tag the interface with vlan 10 for in and out traffic from vlan 10

Show ip interface brief



You configured **switchport trunk allowed vlan add 10** *on an interface, but VLAN10 doesn't appear in the* **Vlans allowed and active in management domain** *section of the* **show interfaces trunk** *command output. What might be the reason?*

a) VLAN10 doesn't exist on the switch.



If the target pc is in the same vlan and same subnet then no inter vlan routing

Native VLAN on a router (ROAS)

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

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

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

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

(OR)

```
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)#
```

```
!
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
!
```

Encapsulation dot1Q 100 native gives native vlan in router

Layer 3 (Multilayer) Switches

Layer 2 switch

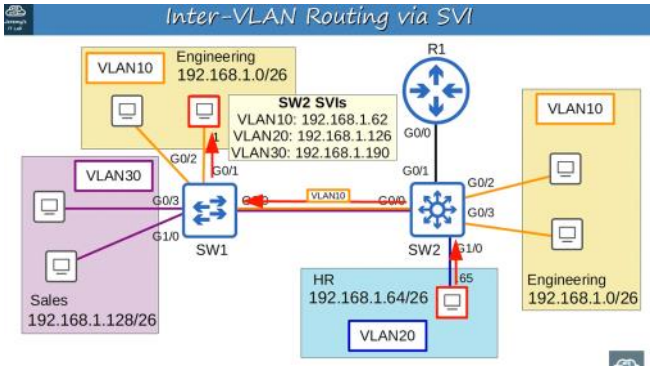
Layer 3 switch

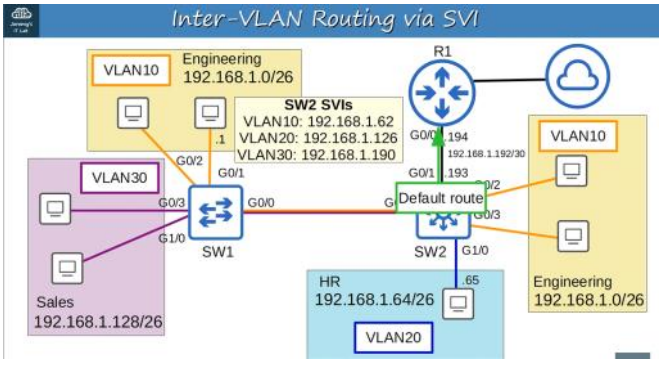
Layer 3 (Multilayer) Switches

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

Inter-VLAN Routing via SVI

- SVIs (Switch Virtual Interfaces) are the virtual interfaces you can assign IP addresses to in a multilayer switch.
- Configure each PC to use the SVI (NOT the router) 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.





R1(config)# no interface g0/0.10
R1(conifg)# no interface g0/0.20
R1(conifg)# no interface g0/0.30
R1(conifg)# default interface g0/0
R1(conifg)# do show ip interface brief
R1(config)# ip address 192.168.1.194 255.255.255.252

SW2(conifg)# default interface g0/1
SW2(config)# ip routing (layer 3 routing is enabled on switch only if we give this command)
SW2(config)# interface g0/1
SW2(config-if)# no switchport(configures the interface as a routed port)
SW2(config-if)# ip address 192.168.1.193 255.255.255.252
SW2(config-if)# do show ip interface brief

Default to Router to internet:

SW2(config)# ip route 0.0.0.0 0.0.0.0 192.168.1.194
SW2(config)# do show ip route
SW2# show interfaces status
(here the particular interface we configured as a routing port will show vlan as 'routed')

SVI configuration:

```
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
```

(interface vlan 10 or interface vlan10))
SVI's are shutdown by default so give no shutdown

1) The VLAN must exist on the switch.
2) 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.
3) The VLAN must not be shutdown (you can use the **shutdown** command to disable a VLAN.
4) The SVI must not be shutdown (SVIs are disabled by default)

(the default gateway address in pc's is now the vlan's ip address

(EXSIM-MAX PRACTICE EXAMS)
AT LAST AFTER FINISHING THE COURSES WE CAN SEE THE BOSON NETSIM PRACTICE LABS

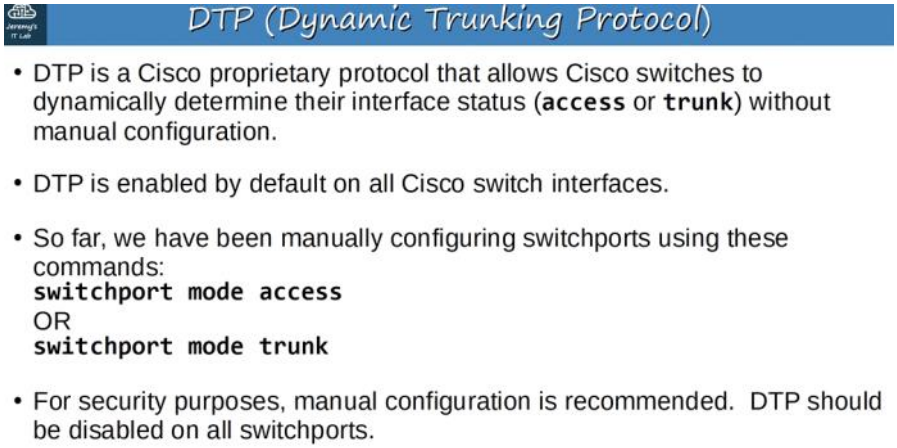
When we remove the sub interfaces we created in router in original cisco devices it will be in the entry and the status will show deleted, also when enabling ip routing the routes should show both connected and local route(in packet tracer might be showing only connected route)

DTP AND VTP PROTOCOL

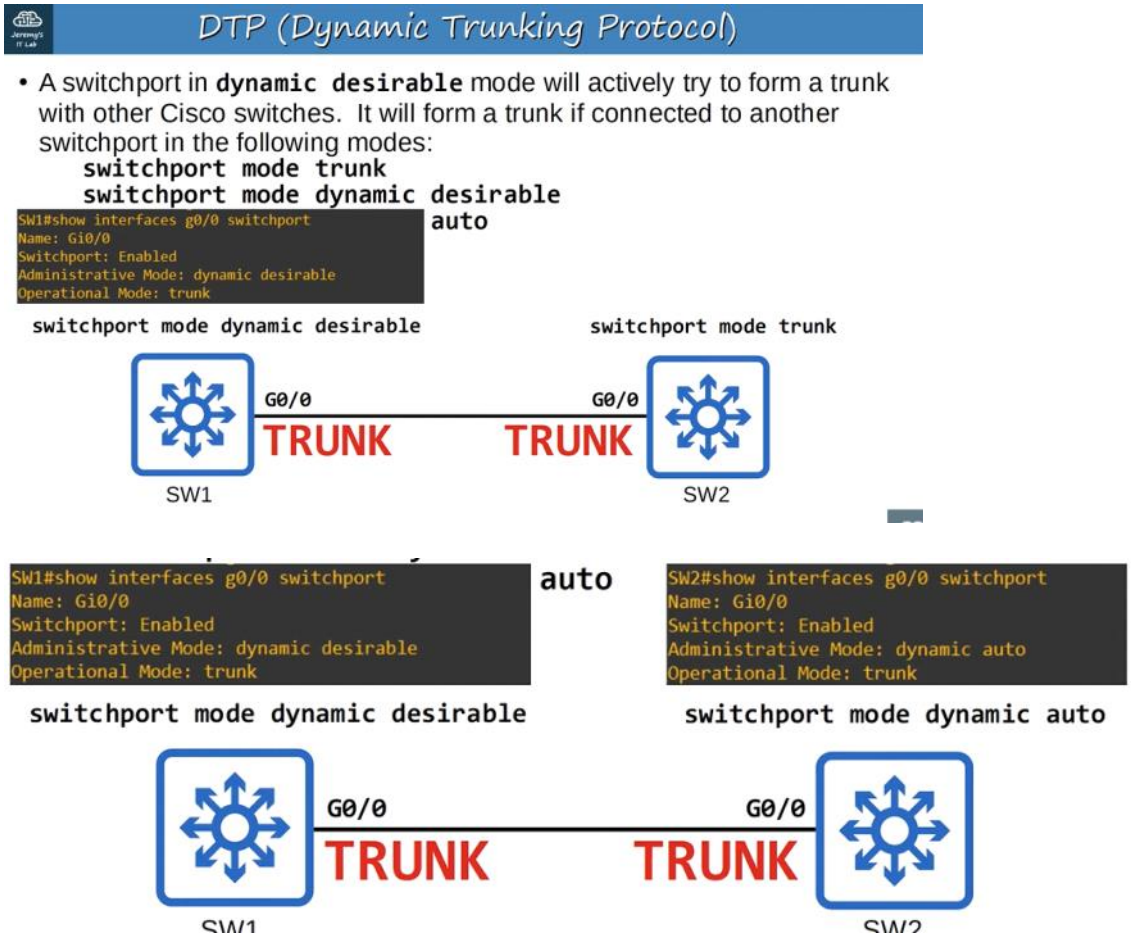
Wednesday, February 8, 2023 2:48 PM

DTP(dynamic trunking protocol),VTP(VLAN trunking protocol)

- These are cisco proprietory so wont work in any other devices



Switchport mode dynamic auto/desirable (for DTP functionality



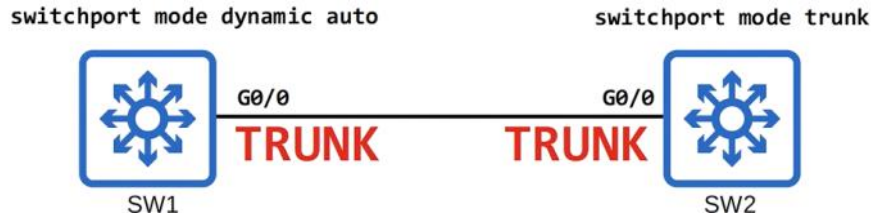
- A switchport in **dynam**
with other Cisco switc
switchport in the follo
switchport mod
switchport mod

```
SW1#show interfaces g0/0 switc
Name: Gi0/0
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: static access
```

'static access' means an access port that belongs to a single VLAN that doesn't change (unless you configure a different VLAN).

There are also 'dynamic access' ports, in which a server automatically assigns the VLAN depending on the MAC address of the connected device.
(this is out of the scope of the CCNA)

- A switchport in **dynamic auto** mode will NOT actively try to form a trunk with other Cisco switches, however it will form a trunk if the switch connected to it 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



(this resolves to an error) up

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 (Dynamic Trunking Protocol)

- On older switches, **`switchport mode dynamic desirable`** is the default administrative mode.
- On newer switches, **`switchport mode dynamic auto`** is the default administrative mode.
- You can disable DTP negotiation on an interface with this command: **`switchport nonegotiate`**
- Configuring an access port with **`switchport mode access`** also disables DTP negotiation on an interface.
- It is recommended that you disable DTP on all switchports and manually configure them as access or trunk ports.
- Switches that support both **802.1Q** and **ISL** trunk encapsulations can use DTP to negotiate the encapsulation they will use.
- This negotiation is enabled by default, as the default trunk encapsulation mode is: **`switchport trunk encapsulation negotiate`**
- **ISL** is favored over **802.1Q**, so if both switches support ISL it will be selected.
- DTP frames are sent in VLAN1 when using **ISL**, or in the native VLAN when using **802.1Q** (the default native VLAN is VLAN1, however).

```
SW1(config-if)#switchport mode dynamic desirable
SW1(config-if)#do show interfaces g0/0 switchport
Name: Gi0/0
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: isl
Negotiation of Trunking: On
```

```
SW2(config-if)#switchport mode dynamic desirable
SW2(config-if)#do show interfaces g0/0 switchport
Name: Gi0/0
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: isl
Negotiation of Trunking: On
```

VTP(VLAN trunking protocol)

VTP (VLAN Trunking Protocol)

- VTP allows you to configure VLANs on a central VTP server switch, and other switches (VTP clients) will synchronize their VLAN database to the server.
- It is designed for large networks with many VLANs, so that you don't have to configure each VLAN on every switch.
- It is rarely used, and it is recommended that you do not use it.
- There are three VTP versions: 1, 2, and 3.
- There are three VTP modes: **server**, **client**, and **transparent**.
- Cisco switches operate in VTP server mode by default.

VTP (VLAN Trunking Protocol)

- VTP Servers:
  Can add/modify/delete VLANs.
  Store the VLAN database in non-volatile RAM (NVRAM).
  Will increase the **revision number** every time a VLAN is added/modified/deleted.
  Will advertise the latest version of the VLAN database on trunk interfaces, and the VTP clients will synchronize their VLAN database to it.
  VTP servers also function as VTP clients
  Therefore, a VTP server will synchronize to another VTP server with a higher revision number.

- VTP clients:
  Cannot add/modify/delete VLANs.
  Do not store the VLAN database in NVRAM. **(in VTPv3, they do)**
  Will synchronize their VLAN database to the server with the highest revision number in their VTP domain.

Will advertise their VLAN database, and forward VTP advertisements to other clients over their trunk ports.

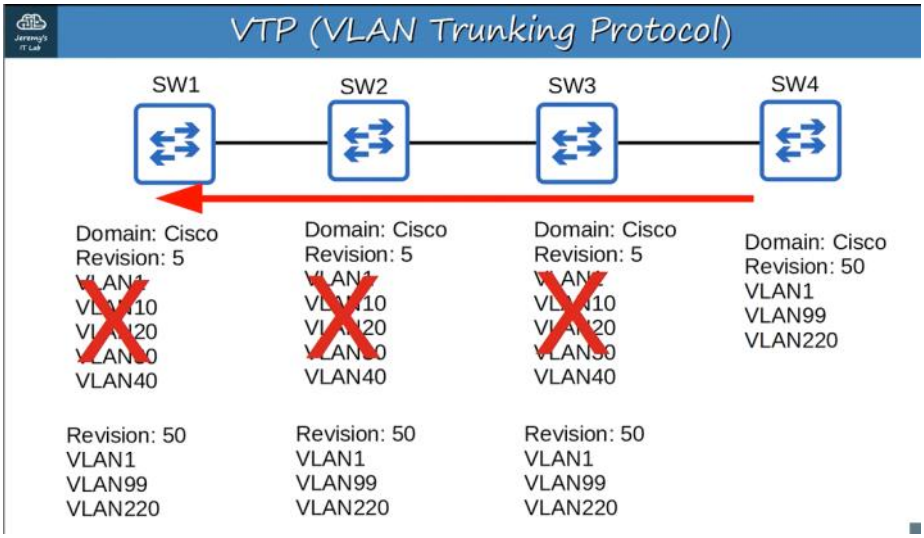
SW1# show vtp status
(gives the details about vtp)

SW1(config)# vtp domain cisco
SW1(config)# vlan 10
SW1(config-vlan)# name engineering

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.

One danger of VTP:
If you connect an old switch with a higher revision number to your network (and the VTP domain name matches), all switches in the domain will sync their VLAN database to that switch.



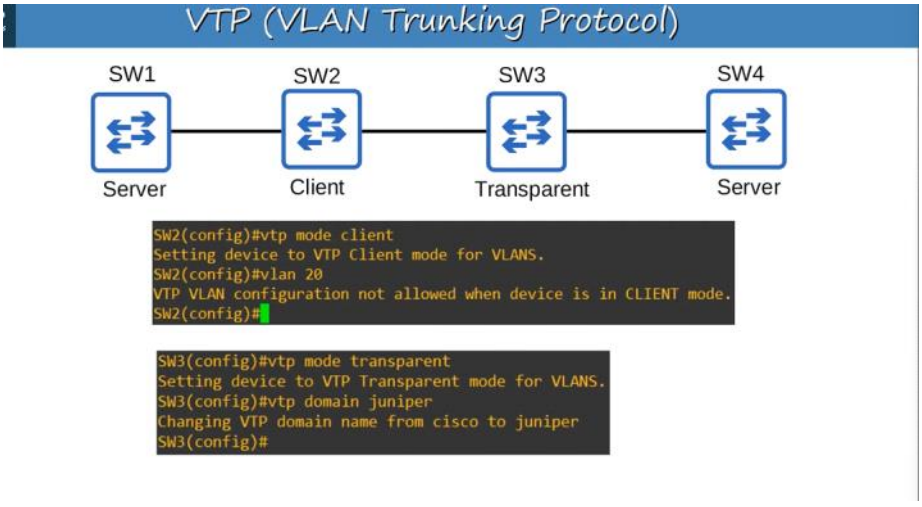
THESE TWO PROTOCOLS ARE NOT USED IN MODERN NETWORKS

- VLAN Transparent mode:

  Does not participate in the VTP domain (does not sync its VLAN database).

  Maintains its own VLAN database in NVRAM. It can add/modify/delete VLANs, but they won't be advertised to other switches.

  Will forward VTP advertisements that are in the same domain as it.




```
SW1(config)#vlan 20
SW1(config-vlan)#name sales
SW1(config-vlan)#exit
SW1(config)#do show vlan brief

VLAN Name                              Status    Po
---- --------------------------------- --------- --
1    default                           active    Gi
                                                 Gi
                                                 Gi
                                                 Gi
10   engineering                       active
20   sales                             active
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup
SW1(config)#
```

```
SW1(config)#do show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0c09.f956.1300
Configuration last modified by 0.0.0.0 at 5-4-20 03:40:01
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
--------------
VTP Operating Mode              : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 7
Configuration Revision          : 4
MD5 digest                      : 0x8F 0x9C 0x81 0x4B 0x5
                                  0xE8 0xA3 0x98 0xFD 0xC
SW1(config)#
```

```
SW2#show vlan brief

VLAN Name                             Status    P
---- -------------------------------- --------- -
1    default                          active    G
                                                G
                                                G
                                                G
10   engineering                      active
20   sales                            active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
SW2#
```

```
SW2#show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0c09.f9ab.0800
Configuration last modified by 0.0.0.0 at 5-4-20 03:40:01

Feature VLAN:
--------------
VTP Operating Mode              : Client
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 7
Configuration Revision          : 4
MD5 digest                      : 0x8F 0x9C 0x81 0x4B 0x
                                  0xE8 0xA3 0x98 0xFD 0x
SW2#
```

```
SW3#show vlan brief

VLAN Name                             Status    P
---- -------------------------------- --------- -
1    default                          active    G
                                                G
                                                G
                                                G
10   engineering                      active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
SW3#
```

```
SW3#show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : juniper
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0c09.f9fa.e700
Configuration last modified by 0.0.0.0 at 5-4-20 03:33:08

Feature VLAN:
--------------
VTP Operating Mode              : Transparent
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 6
Configuration Revision          : 0
MD5 digest                      : 0xDB 0x6A 0xDB 0x61 0x
                                  0x59 0x73 0x4E 0xF4 0x
SW3#
```

Changing the VTP domain to an unused domain will reset the revision number to 0.

Changing the VTP mode to transparent will also reset the revision number to 0.

```
SW4#show vlan brief

VLAN Name                             Status    P
---- -------------------------------- --------- -
1    default                          active    G
                                                G
                                                G
                                                G
10   engineering                      active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
SW4#
```

```
SW4#show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0c09.f972.8700
Configuration last modified by 0.0.0.0 at 5-4-20 03:33:08
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
--------------
VTP Operating Mode              : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 6
Configuration Revision          : 3
MD5 digest                      : 0xFC 0x05 0xC0 0x82 0x
                                  0xF4 0x35 0x5D 0x76 0x
SW4#
```

```
SW3(config)#vtp domain cisco
Changing VTP domain name from juniper to cisco
SW3(config)#
*May  4 04:06:00.101: %SW_VLAN-6-VTP_DOMAIN_NA
SW3(config)#
```

```
SW4#show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 1
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0c09.f972.8700
Configuration last modified by 0.0.0.0 at 5-4-20 04:15:14
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
--------------
VTP Operating Mode              : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 11
Configuration Revision          : 12
MD5 digest                      : 0xDB 0x14 0xEF 0x30 0
                                  0xEC 0x6C 0x96 0xAD 0
SW4#
```

```
SW1(config)#vtp version 2
SW1(config)#do show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 2
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0c09.f956.1300
Configuration last modified by 0.0.0.0 at 5-4-20 04:19:30
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
--------------
VTP Operating Mode              : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 11
Configuration Revision          : 13
MD5 digest                      : 0xE4 0xC9 0x65 0x0A 0
                                  0x99 0xB2 0x16 0x81 0
SW1(config)#
```

```
SW4#show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 2
VTP Domain Name                 : cisco
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0c09.f972.8700
Configuration last modified by 0.0.0.0 at 5-4-20 04:19:30
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
--------------
VTP Operating Mode              : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 11
Configuration Revision          : 13
MD5 digest                      : 0xE4 0xC9 0x65 0x0A 0x
                                  0x99 0xB2 0x16 0x81 0x
SW4#
```

VTP V2 is not much different than VTP V1. The major difference is that VTP V2 introduces support for Token Ring VLANs. If you use Token Ring VLANs, you must enable VTP V2. Otherwise, there is no reason to use VTP V2.

b) Interfaces on old switches default to **`switchport mode dynamic desirable`**

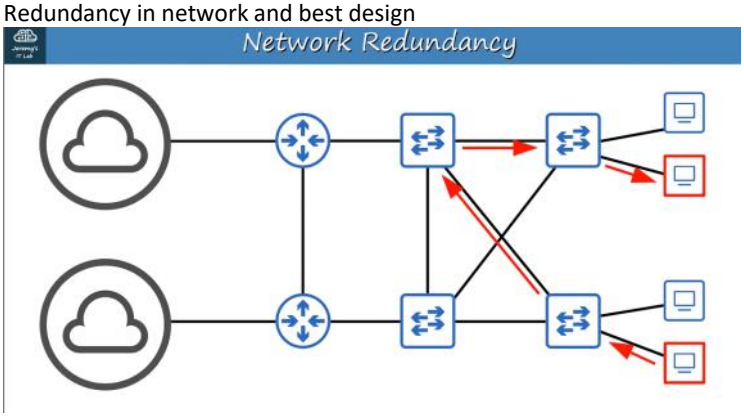
New switches come with dynamic auto by default

CONCEPTS PART4

Wednesday, February 8, 2023 7:56 PM

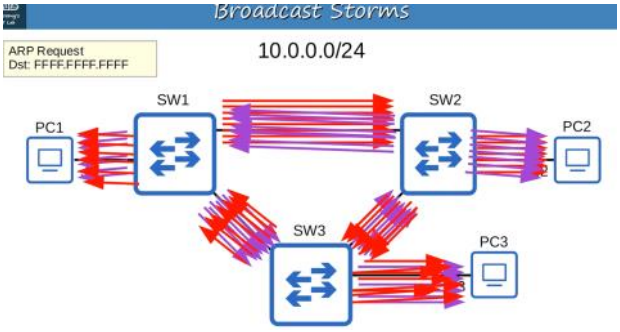
Spanning tree protocol:(layer 2 protocol)

Redundancy in network and best design



We should not have a design where each object is dependent on only one switch or pc or router etc..
Like if a switch is disconnected to the pc the pc should have multiple interfaces in order to have another switch to support it(same applies to router,switch etc..)

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**.



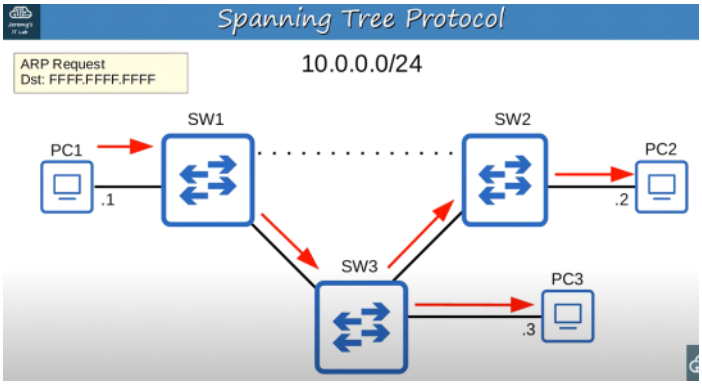
Network congestion isn't the only problem. Each time a frame arrives on a switchport, the switch uses the source MAC address field to 'learn' the MAC address and update its MAC address table. When frames with the same source MAC address repeatedly arrive on different interfaces, the switch is continuously updating the interface in its MAC address table. This is known as **MAC Address Flapping**.

STP:

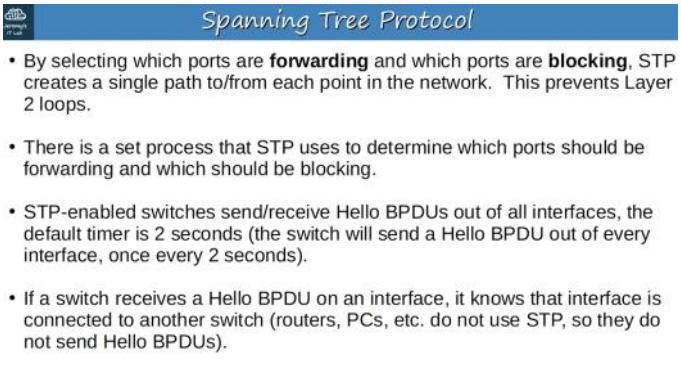
- Classic spanning tree protocol is IEEE 802.1D
- Switches from all vendors run STP by default
- STP prevents layer 2 loops by placing redundant ports in a blocking state,essentially disabling the interface
- These interfaces act as backups that can enter a forwarding state if an active(=currently forwarding)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

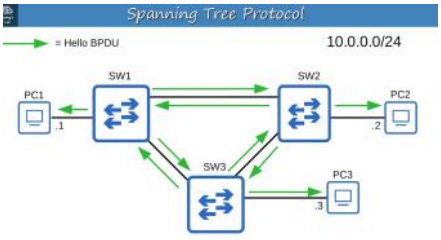
Before the switch device and after the hub device we saw in first day lecture there is a bridge device

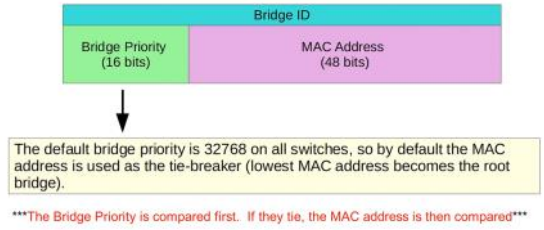
Spanning Tree Protocol still uses the term 'bridge'. However, when we use the term 'bridge', we really mean 'switch'. Bridges are not used in modern networks.

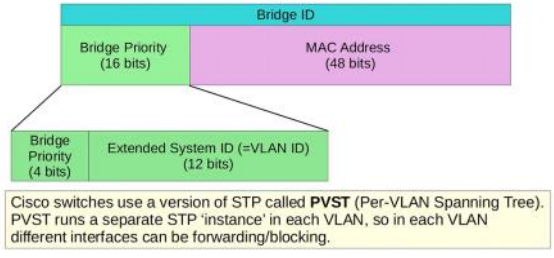


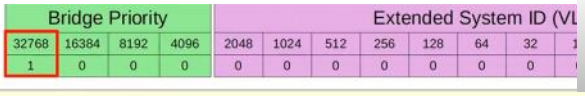
In the above diagram in short when broadcast frames are sent it will block the traffic in a interface given as dotted lines in sw1 and floods it into another interface in forwarding state and no loop occurs and if the other interface fails the switches will adjust and enable the forwarding state in the previous interface and disable this other interface and flood the frames so no loop occurs

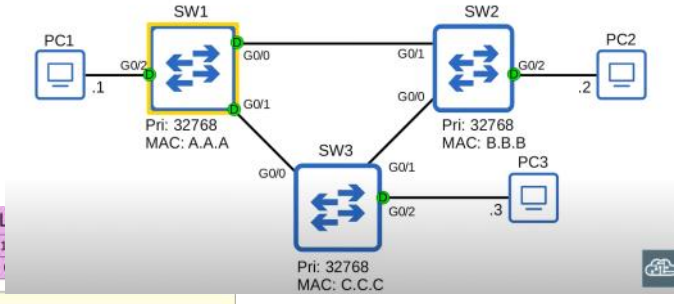












In the default VLAN of 1, the default bridge priority is actually **32769** (32768 + 1).

Bridge Priority				Extended System ID (VL							
32768	16384	8192	4096	2048	1024	512	256	128	64	32	1
1	0	0	0	0	0	0	0	0	0	0	

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?

The **bridge priority** + **extended system ID** is a single field of the bridge ID, however the extended system ID is set and cannot be changed (because it is determined by the VLAN ID.

Therefore, the you can only change the total bridge priority (bridge priority + extended system ID) in units of 4096, the value of the least significant bit of the bridge priority.

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

All interfaces on the root bridge are **designated ports**.
Designated ports are in a forwarding state.

Spanning Tree Protocol

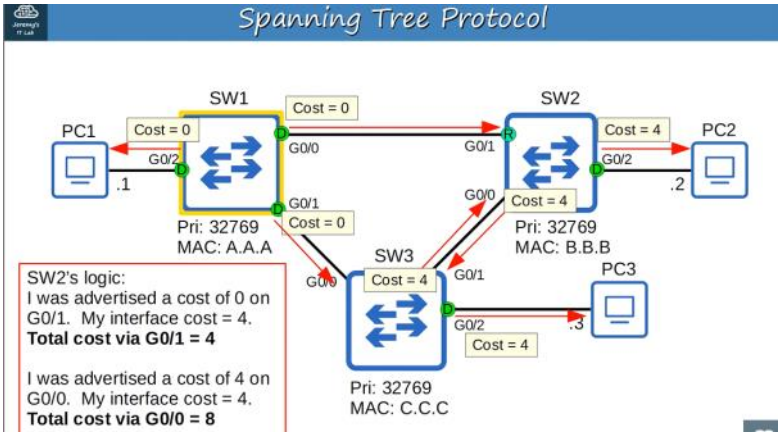
- When a switch is powered on, it assumes it is the root bridge.
- It will only give up its position if it receives a 'superior' BPDU (lower bridge ID).
- Once the topology has converged and all switches agree on the root bridge, only the root bridge sends BPDUs.
- Other switches in the network will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

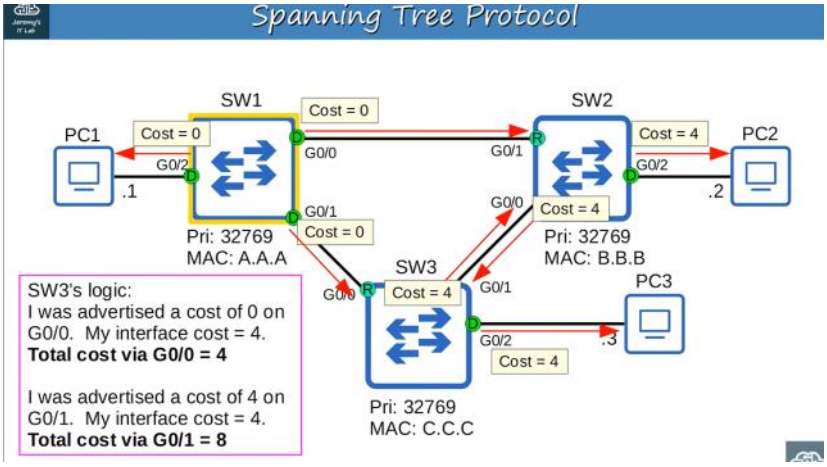
STP:

1) The switch with the lowest bridge ID is elected as the root bridge. All ports on the root bridge are **designated ports** (forwarding state).

2) Each remaining switch will select ONE of its interfaces to be its **root port.** The interface with the lowest *root cost* will be the root port. Root ports are also in a forwarding state.

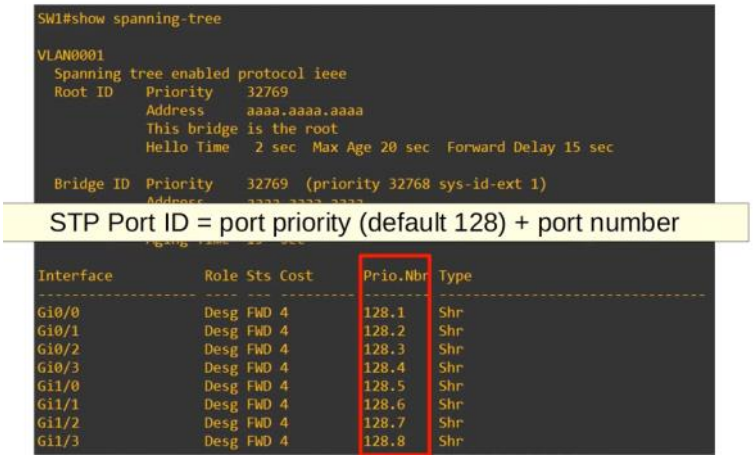
Speed	STP Cost
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2



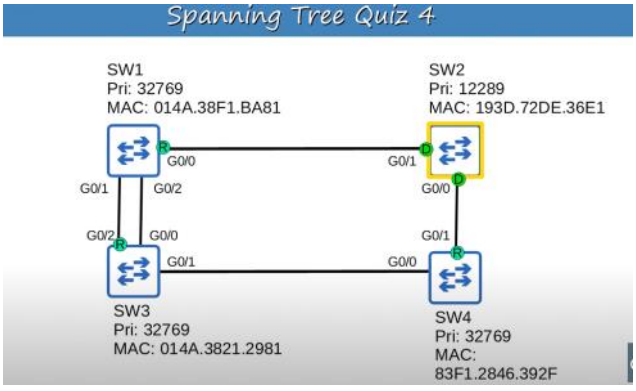


If both interfaces have same root cost then it sees the lowest bridge id to determine root port

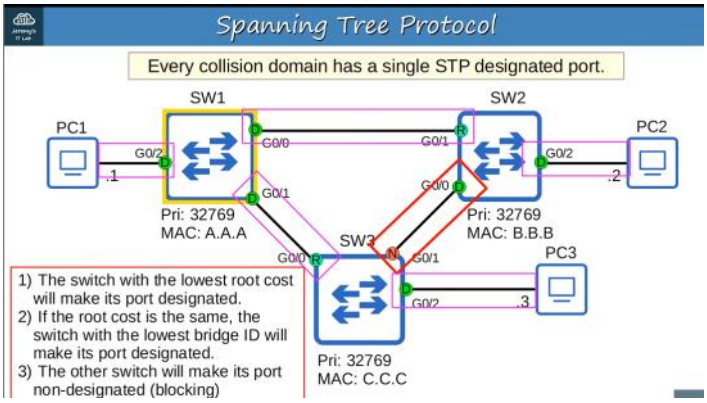
SW1# show spanning-tree (shows the spanning tree status for interfaces)

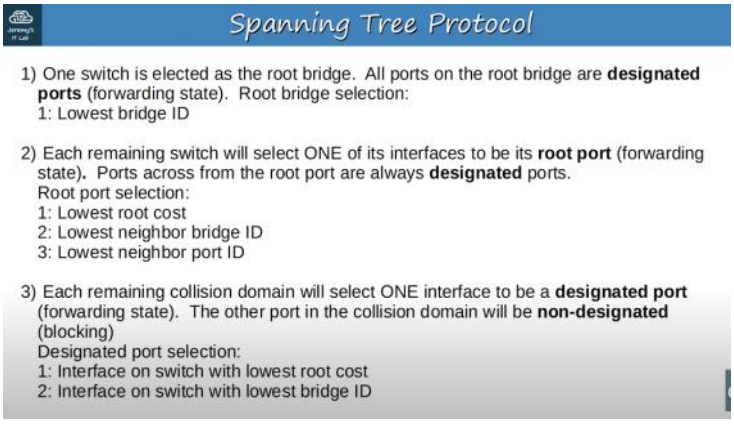


If the both switches has two connections and cost id is same ,bridge id is same then port id is taken into account



In the above diagram the root port is g0/2 in sw3 since the neighbouring interface has the least port id

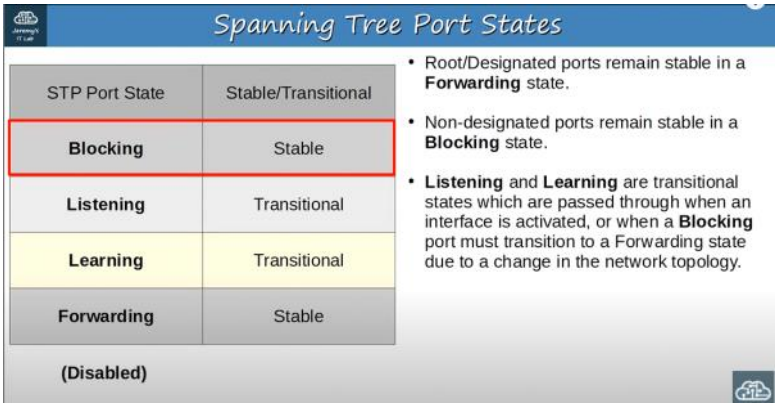




We need to find the root cost till the path end with root switch, when seeing the nondesignated port if the oppsite is root switch then that interface is non designated

In packet tracer Options>preferences>show link lights check box to see which ones are designated

SW3# show spanning-tree (we can add 'vlan 10' if we want to see for particular vlan)
SW3# show spanning-tree detail
SW3# show spanning-tree summary(about how many are in locking listening etc..)



Blocking -> do not send,receive regular traffic,do not forward STP BPDU's,do not learn MAC address,receive BPDU
Listening ->only designated or root ports enter the listening state,15 sec long by default,only forwards/receives STP BPDU's
does not learn MAC addresses,does not send/receive traffic
After listening learning state occurs
Learning -> difference between listening,learning is that learning learns mac addresses
Forwarding -> send/receive BPDUS normal traffic,learns mac addreses

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 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)

Hello :

- Forwards the bpdu's to the other switches to the root ports and the other switches send them to other using designated ports

STP Timer	Purpose	Duration
Max Age	How long an interface will wait to change the STP topology after ceasing to receive Hello BDPUs. The timer is reset every time a BPDU is received.	20sec (10* hello)

- If another BPDU is received before the 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 reevaluate its STP choices, including root bridge, and 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 a total of **50 seconds** for a blocking interface to transition to forwarding.
- These timers and transitional states are to make sure that loops aren't accidentally created by an interface moving to forwarding state too soon.

PVST(per vlan spanning tree) => only supports trunk encapsulation
PVST+ => supports 802.1Q

Regular STP(not cisco PVST+) uses a destination MAC address of 0180.c200.0000

PortFast:

- Actually by default the ports go through a listening and learning state in order to forward packets but that will take 30 seconds(listening+learning)
- The portfast feature in switch allows to directly fo to forwarding state and not to wait till the two phases
- Only should do with pc as the end hosts for that interface

SW1(config)#Interface g0/2
SW1(config-if)#spanning-tree portfast

- Only have effect when the interface is in non trunking (access) mode

SW1(config)# spanning-tree portfast default
(on all access ports)

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

```
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.

Shitdown,no shutdown to disable bpduguard
(only works if the issue with us is fixed otherwise will be up again

(only should enable in interface connecting to pc if it is connected to switch then we need to remove otherwise connections w ont occur between that switch and this switch)

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.

```
SW1(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 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
```

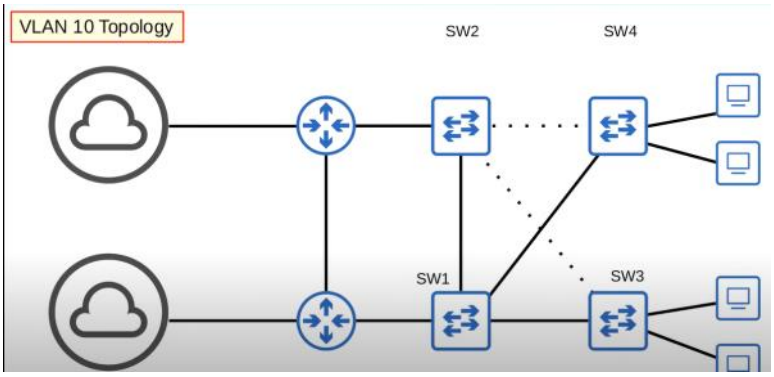
The **spanning-tree vlan** *vlan-number* **root primary** command sets the STP priority to 24576. If another switch already has a priority lower than 24576, it sets this switch's priority to 4096 less than the other switch's priority.

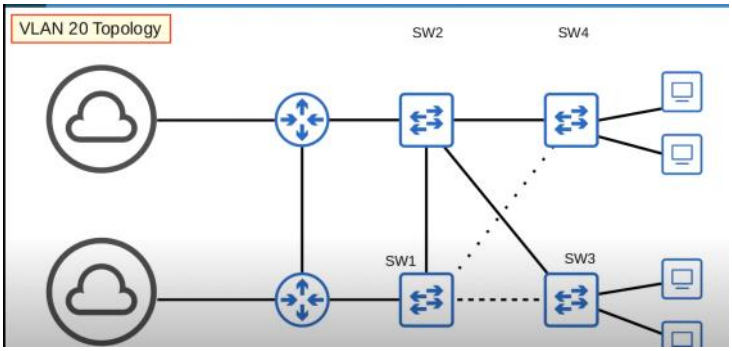
Spanning-tree vlan 1 priority 24576 (to manually set priority)

```
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** *vlan-number* **root secondary** command sets the STP priority to 28672.

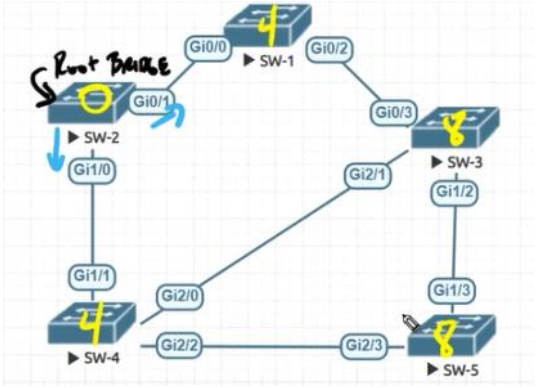
LOOAD BALANCING BETWEEN VLANS





Spanning-tree vlan 1 cost/port-priority
(200)/(in increments of 32)

STP port ID of 0x8002 => 80 in decimal => 128 is the port priority



Here we need to caculate the cost from the root bridge so

SW2(g0/1)=0 + SW1(G0/0)=4 => SW1(G/0/0)=4
SW1(g0/2)=(4 is cost already came to switch so 4 will be forwarding) + SW3(G0/3)=4 => SW1(G/0/3)=8
SW3(g0/3)=(already this switch received the cost of 8 total so it will give the cost to g1/2 interface so cost is 8) + SW5(G1/3)=4 => SW5(G/1/3)=12

(in short for a switch an interface from which when travelling to the network to reach root bridge has the lowest cost it will be root port)

Comparison of STP versions:

Industry Standards(IEEE)	Cisco versions
Spanning tree protocol(802.1D)-classic All vlan's share one STP instance so cant load balance (upto 50 seconds wait time for oth)	Per VLAN Spanning Tree Plus(PVST+) Each vlan has its own STP instance Load balance by blocking different ports in each vlan
Rapid Spanning Tree(802.1w) Much faster and here also all vlans share one STP instance,cant load balance	Rapid PVST+ Cisco's upgrade to 802.1w Same as above but faster
Multiple spanning tree protocol(802.1s) Uses modified Rapid STP mechanics Can group multiple VLANs into different instances to perform load balancing	

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

10TBPS =>2

Rapid Spanning Tree Port States

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

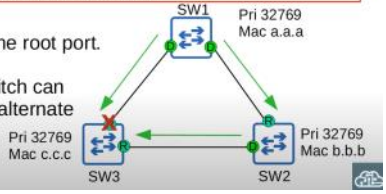
- If a port is administratively disabled (**`shutdown`** command) = discarding state
- If a port is enabled but blocking traffic to prevent Layer 2 loops = discarding state

Rapid Spanning Tree Port Roles

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

RSTP: Alternate port role

- The RSTP **alternate** port role is a discarding port that receives a superior BPDU from another switch.
- 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+.
- Functions as a backup to the root port.
- If the root port fails, the switch can immediately move its best alternate port to forwarding.

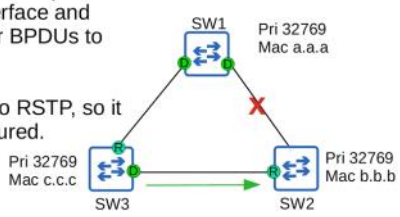


RSTP: BackboneFast functionality

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

BackboneFast allows SW3 to expire the made age timers on its interface and rapidly forward the superior BPDUs to SW2.

This functionality is built into RSTP, so it does not need to be configured.

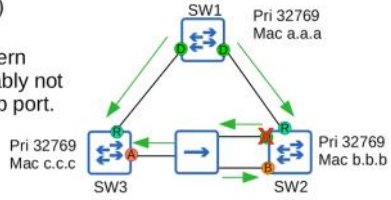


UplinkFast / BackboneFast Summary

- **UplinkFast** and **BackboneFast** are two optional features in classic STP. They must be configured to operate on the switch (not necessary to know for 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).
- If you want to learn more, do a Google search for 'spanning tree uplinkfast' or 'spanning tree backbonefast'.

RSTP: Backup port role

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

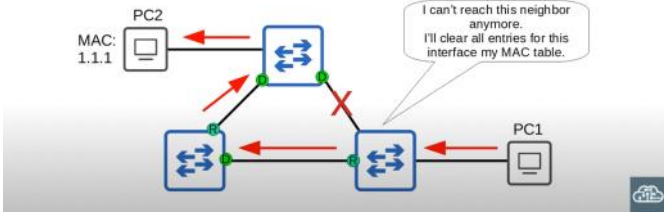


Rapid STP is compatible with Classic STP. The interface(s) on the Rapid STP-enabled switch connected to the Classic STP-enabled switch will operate in Classic STP mode (timers, blocking → listening → learning → forwarding process, etc).

Protocol version for STP is 0 and RSTP is 2

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 neighbor lost if it misses 3 BPDUs (6 seconds). It will then 'flush' all MAC addresses learned on that interface.

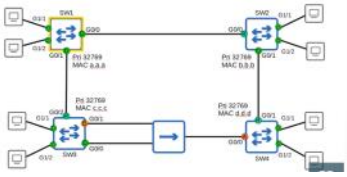


- RSTP distinguishes between three different 'link types'.
- **Edge**: a port that is connected to an end host. Moves directly to forwarding, without negotiation.
- **Point-to-point**: a direct connection between two switches.
- **Shared**: a connection to a hub. Must operate in half-duplex mode.

RSTP Link Types: Edge

- Edge ports are 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.

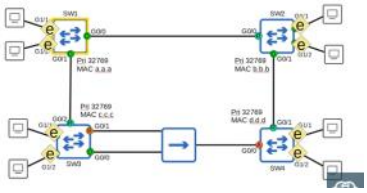
```
SW1(config-if)# spanning-tree portfast
```



RSTP Link Types: Point-to-Point

- 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).

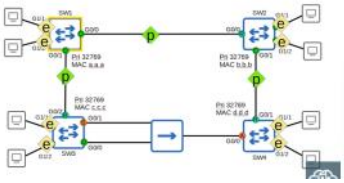
```
SW1(config-if)# spanning-tree link-type point-to-point
```



RSTP Link Types: Shared

- 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).

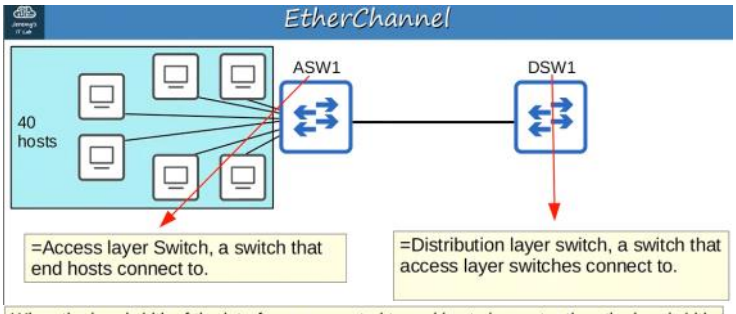
```
SW1(config-if)# spanning-tree link-type shared
```



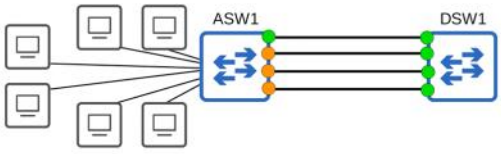
- Rapid PVST+
  - → RSTP port states (discarding, learning, forwarding)
  - → RSTP port roles (root, designated, alternate, backup)
  - → STP optional features built into in RSTP (UplinkFast, BackboneFast, PortFast)
  - → RSTP BPDU (sent by all switches, not just the root bridge)
  - → RSTP link types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

CONCEPTS PART5

Saturday, February 11, 2023 3:09 PM

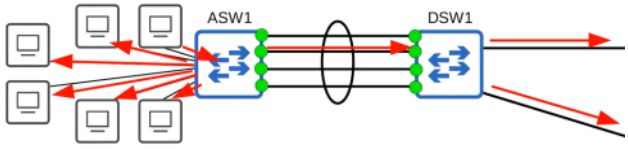


When the bandwidth of the interfaces connected to end hosts is greater than the bandwidth of the connection to the distribution switch(es), this is called **oversubscription**. Some oversubscription is acceptable, but too much will cause congestion.



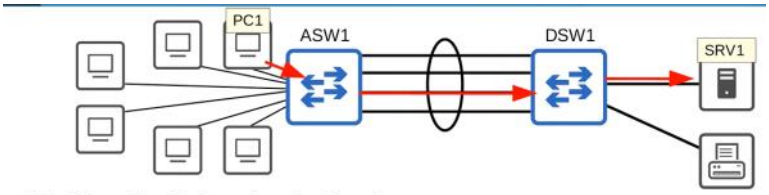
- If you connect two switches together with multiple links, all except one will be disabled by spanning tree.
- If all of ASW1's interfaces were forwarding, Layer 2 loops would form between ASW1 and DSW1, leading to broadcast storms.
- Other links will be unused unless the active link fails. In that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

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



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

- Some other names for an EtherChannel are:
  - Port Channel
  - LAG (Link Aggregation Group)



- 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, which can cause problems.

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

ASW1# show etherchannel load-balance

(shows the configuration like which input is used for intreface selection and etc..)
ASW1(config)# port-channel load-balance src-dst-mac
(changes the input used for selection)

```
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
```

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

PAgP Configuration:
ASW1(config)# interface range g0/0 -3
ASW1(config-if-range)# channel-group 1 mode desirable(active,auto,on,passive)

Auto,desirable = pagp
On = static
Active,passive = LACP

(PAGP)

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

LACP

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

STATIC

On mode only works with **on** mode (**on** + **desirable** or **on** + **active** will not work)

The channel-group number has to match for member interfaces on the same switch.
However, it **doesn't** have to match the channel-group number on the other switch.
(**channel-group 1** on ASW1 can form an EtherChannel with **channel-group 2** on DSW1)

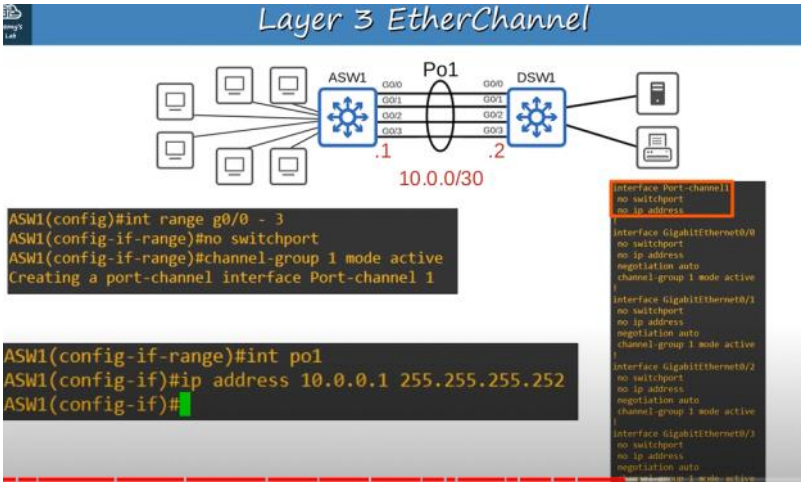
To manually configure the channel-protocol use below
ASW1(config-if-range)# channel-protocol ?(lacp,pagp)

```
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
```

- 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.

ASW1# show etherchannel summary
(to get the ether/port channels created and also have some labels tagged to the portchannel,ports name that can be defined using the detail of each letter)

ASW1# show etherchannel port-channel(to see number of ports,protocol used,state of ports etc..)

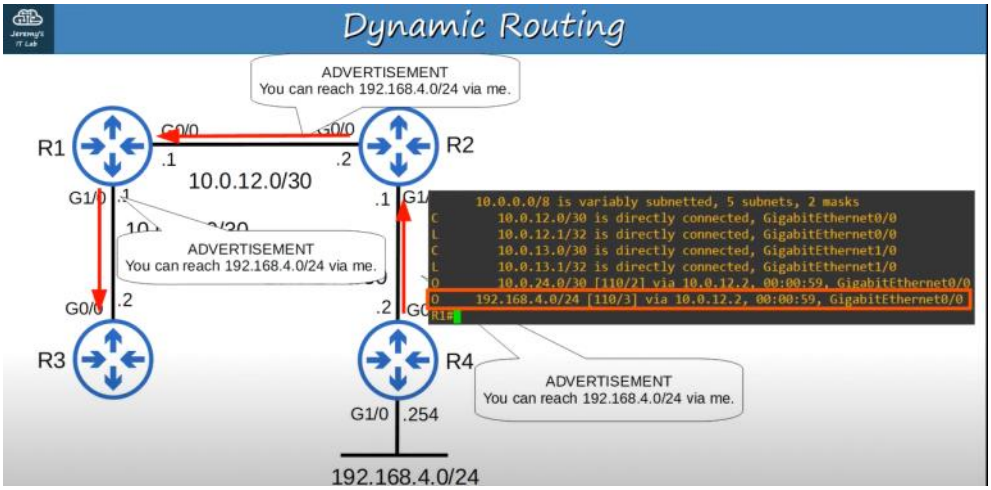



```
SW(config) port-channel load-balance mode
 #configures the EtherChannel load-balancing method on the switch

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

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

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

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

When manualy confugring channel protocol if we give channel group mode which has different protocol the command is rejected

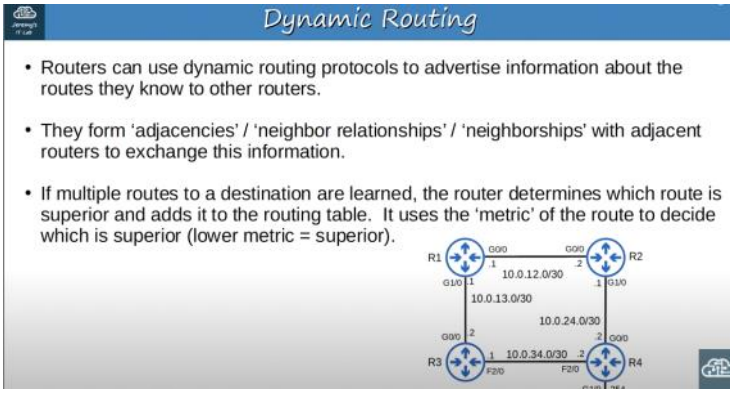
Network route : A route to a network/subnet with masklength < /32 (usually connected route in routers)

Host route: a route to a specific host (usually a local route in router)



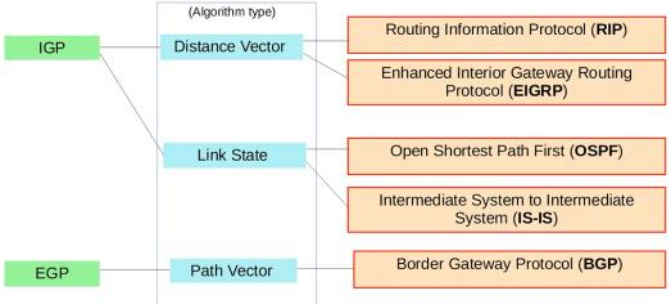
If an interface goes down then all other routers will delete the corresponding routes from its routing table

Dynamic routing will also use similar protocol like spanning tree protocol to determine the best path to reach the destination



Dynamic routing protocols:

- IGP:(OSPF)
  - Interior gateway protocol
  - Used to share routes within a single autonomous system(AS), which is a single organization
- EGP: (BGP)
  - Exterior gateway protocol
  - Used to share routes between different autonomous systems

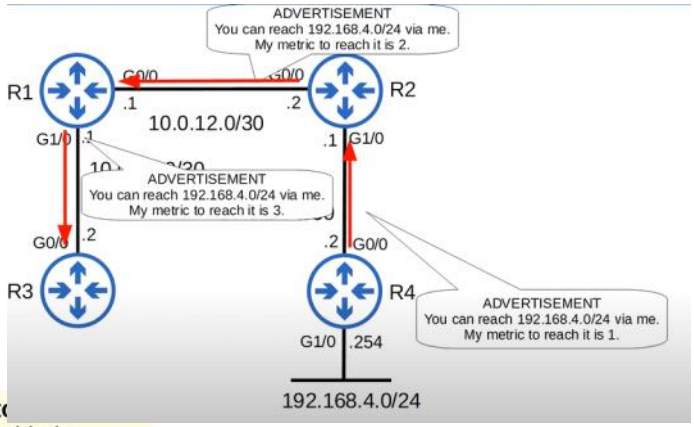


Distance Vector Routing Protocols

- Distance vector protocols were invented before link state protocols.
- Early examples are **RIPv1** and Cisco's proprietary protocol **IGRP** (which was updated to **EIGRP**)
- Distance vector protocols operate by sending the following to their directly connected 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'
- This is because the router doesn't know about the network beyond its neighbors. It only knows the information that its neighbors tell it.
- Called 'distance vector' because the routers only learn the 'distance' (metric) and 'vector' (direction, the next-hop router) of each route.

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 neighbors. These advertisements are passed along to 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 distance vector protocols.



If a router learns two (or more) routes via the **same routing prot**
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.

The above is refered as ECMP(equal cost multi path)
In the routing table the route is displayed with O which referrs as OSPF and in the route it will will be mentioned as
O 192.168.4.0/24 [110/3] via …..
(administrative distance)/(metric cost)

0 metric cost for static routes
1 ad for static routes

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.

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. For example, 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.
- For example, an OSPF route to 192.168.4.0/24 might have a metric of 30, while an EIGRP route to the same destination might have 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).

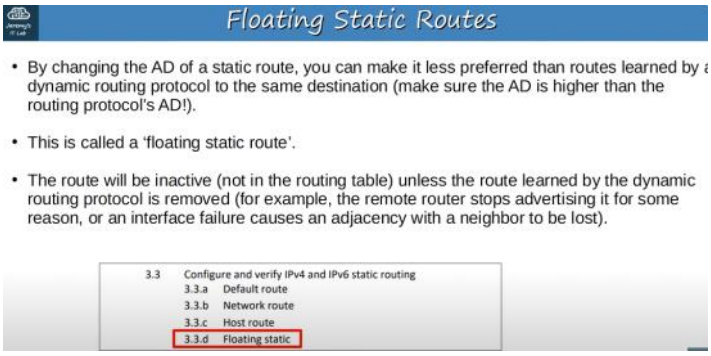
Administrative Distance

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

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.

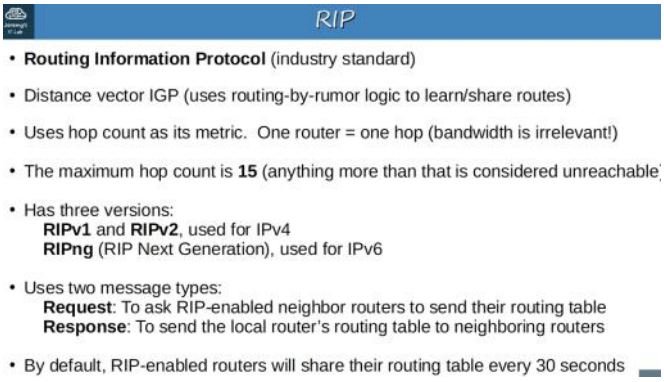
R1(config)# ip route 10.0.0.0 255.0.0.0 10.0.13.2 100(AD) - to statically confugre ad

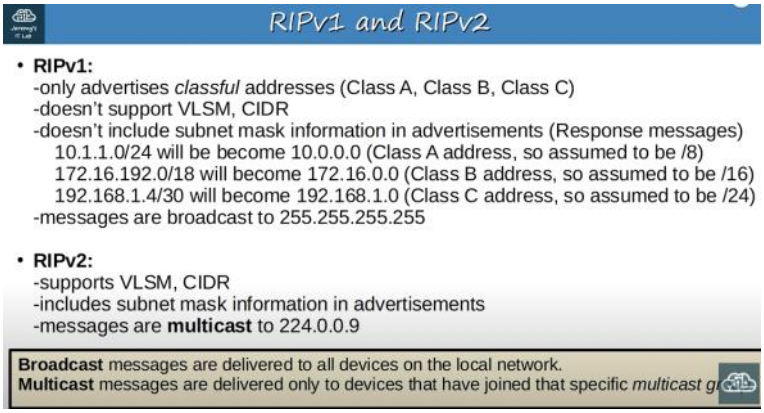


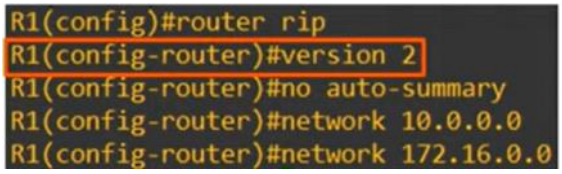
The router will use most specific route to send the packet to the destination
AD is used only to determine which route is placed in routing table when multiple routes to a destination are known router considers most speicific route(longest prefix)

If we are giving default route to a router then that router should have that specific ip as a loopback interface in it to accept it in packet tracer

When we have a lower AD route in the routing table if we give the floating static route then only if the particular interface in which the dynamic route is going is down then only the satic route will be added







The network command is classful, it will automatically convert into a classful networks
Eg: 10.0.12.0 will be converted to 10.0.0.0/8 (no need to enter the network mask)

- This command looks for interfaces with an IP address that is specified

- Activate RIP on the interfaces that fall in this range
- Form adjacencies with connected neighbours
- Advertise the network prefix of the interface(the subnet ip of the link between interfaces)

R1(config-router)#passive-interface g0/0 - to tell the router/rip to not send to this interface the advertisements for rip protocol (OSPF also has same passiveinterface functionality)

After we configure the default route(gateway of last resort) to the internet in this router we want this info to be available in all other routers so the below command will do it
R1(config-router)# default-information originate
(all other routers put routes in their routing table for this)

R1# show ip protocols
(shows the current routing protocol used and its informations)

R1(config-router)# maximum-paths 8 - (maximum only 8 paths can be added to routing table that has the same destination)(same for eigrp,ospf)

```
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)
```

R1(config-router)# distance 85(changing the administrative distance)(same for eigrpp,ospf)

EIGRP(enhanced interior gateway routing protocol)

- Cisco proprietory (but also publicly available for vendors)
- Faster than RIP
- Does not have 15 hop count limit of RIP
- Send messages using multicast address 224.0.0.10
- Is the only IGP that can perform unequal cost load balancing (by default peforms ECMP load balancing over 4 paths like RIP)

RIPv1 broadcasts messages
RIPv2 multicast them to 224.0.0.9
EIGRP multicast them to 224.0.0.10

```
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
```

- 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 it's enabled, disable it.
- The **network** command will assume a classful address if you don't specify the mask.
- EIGRP uses a *wildcard mask* instead of a regular subnet mask.

Wildcard masks

- A wildcard mask is basically an 'inverted' subnet mask.
- All 1s in the subnet mask are 0 in the equivalent wildcard mask. All 0s in the subnet mask are 1 in the equivalent wildcard mask.

```
11111111 . 11111111 . 11111111 . 00000000
  255    .    255   .    255   .    0

00000000 . 00000000 . 00000000 . 11111111
   0     .     0    .     0    .   255

/24
```

- '0' in the wildcard mask = must match
- '1' in the wildcard mask = don't have to match

R1 G2/0 IP address:

```
10101100 . 00010000 . 00000001 . 00001110
  172    .    16    .    1     .    14
```

EIGRP **network** command:

```
10101100 . 00010000 . 00000001 . 00000000
  172    .    16    .    1     .    0

00000000 . 00000000 . 00000000 . 00001111
   0     .     0    .     0    .   15
```

```
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
```

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

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

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

AD is 90 for internal routes and 170 for external 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
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
```

R1 G1/0 IP address:

		172			.		20			.		20			.	17
1	0	1	0	1	1	0	0	.	0	0	0	1	0	0	.	0	0	0	1	0	1	0	0	.	0	0	0	1	0	0	0	1

R1 G2/0 IP address:

		172			.		26			.		20			.	12
1	0	1	0	1	1	0	0	.	0	0	0	1	1	0	.	0	0	0	1	0	1	0	0	.	0	0	0	0	1	1	0	0

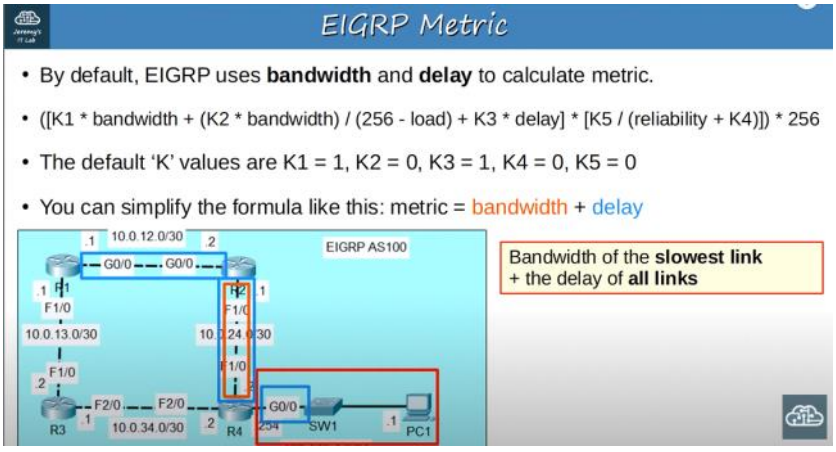
EIGRP **network** command:

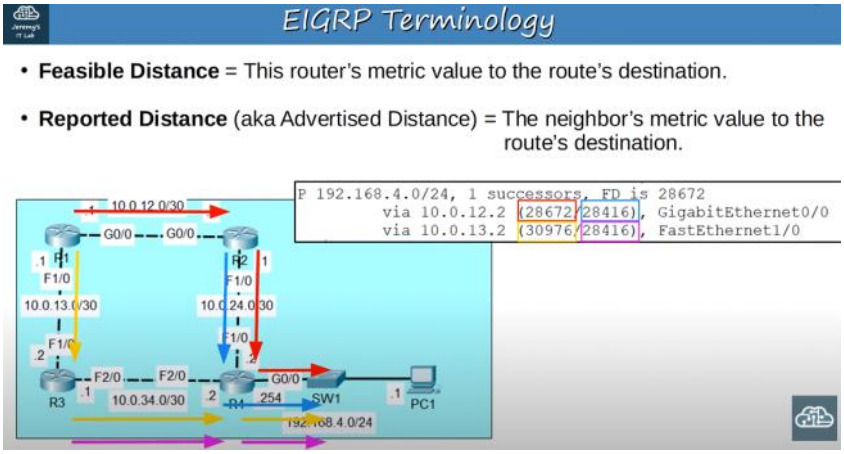
1	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	.	0	0	0	0	0	0	0	0
		128			.		0			.		0			.	0
0	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1
		127			.		255			.		255			.	255

R1(config)# interface loopback 0(0-3431142(somehting))
R1(config-if)# ip address 1.1.1.1 255.255.255.255

Steps:

- Configure loopback interfaces in all routers
- Configure the eigrp in the interfaces(all routers)
  Router eigrp 100
  no auto-summary
  Network 0.0.0.0 255.255.255.255
  Network 1.1.1.1 0.0.0.0
  Passive-interface l0 (loopback 0 interface)
- R1(config-router)# do show ip eigrp neighbord (to confirm the neighbors)
- Do show ip route eigrp (to see routes are added correctly)
- Do show ip eigrp topology (all info)

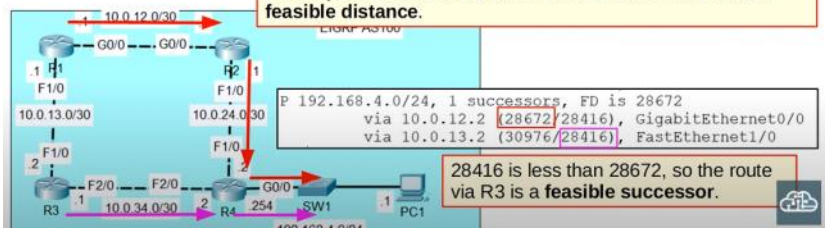




EIGRP Terminology

- **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**.




```
P 192.168.4.0/24, 1 successors, FD is 28672
        via 10.0.12.2 (28672/28416), GigabitEthernet0/0
        via 10.0.13.2 (30976/28416), FastEthernet1/0
```

28416 is less than 28672, so the route via R3 is a **feasible successor**.

EIGRP Unequal-Cost Load-Balancing

```
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
```

Variance 1 = only ECMP load-balancing will be performed

```
P 192.168.4.0/24, 1 successors, FD is 28672
        via 10.0.12.2 (28672/28416), GigabitEthernet0/0
        via 10.0.13.2 (30976/28416), FastEthernet1/0
```

EIGRP Unequal-Cost Load-Balancing

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

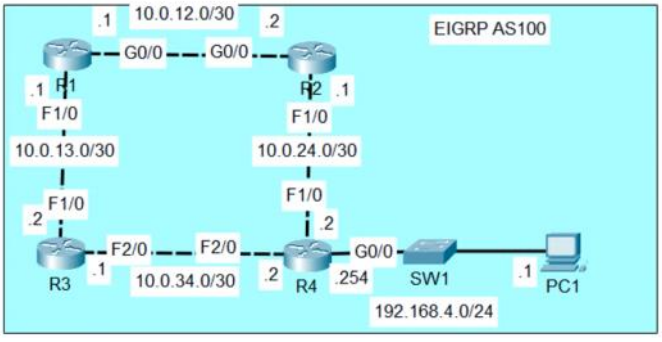
Variance 2 = **feasible successor** routes with an FD up to 2x the **successor** route's FD can be used to load-balance.

EIGRP will only perform unequal-cost load-balancing over **feasible successor** routes. If a route doesn't meet the feasibility requirement, it will NEVER be selected for load-balancing, regardless of the **variance**.

```
P 192.168.4.0/24, 1 successors, FD is 28672
        via 10.0.12.2 (28672/28416), GigabitEthernet0/0
        via 10.0.13.2 (30976/28416), FastEthernet1/0
```

28672 * 2 = 57344
30976 is less than 57344, so the route via R3 can now be used for load-balancing.

```
D    192.168.4.0/24 [90/28672] via 10.0.12.2, 00:11:21, GigabitEthernet0/0
                    [90/30976] via 10.0.13.2, 00:11:21, FastEthernet1/0
```



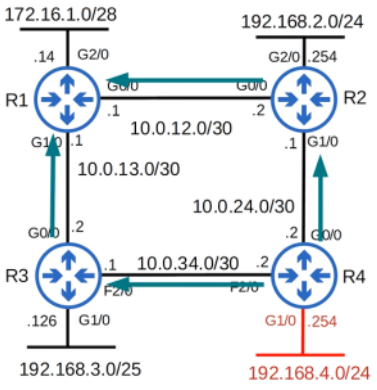
Link State Routing Protocols

- 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 neighbors. These advertisements are passed along to 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 distance vector protocols.

OSPF

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

LSA Flooding



- OSPF is enabled on R4's G1/0 interface.
- R4 creates an LSA to tell its neighbors about the network on G1/0.
- The LSA is flooded throughout the network until all routers have received it.
- This results in all routers sharing the same LSDB.
- Each router then uses the SPF algorithm to calculate its best route to 192.168.4.0/24.

LSA
RID: 4.4.4.4
IP: 192.168.4.0/24
Cost: 1

LSDB

LSA	LSA	LSA	LSA
LSA	LSA	LSA	LSA

Each LSA has an aging timer (30 min by default). The LSA will be flooded again after the timer expires.

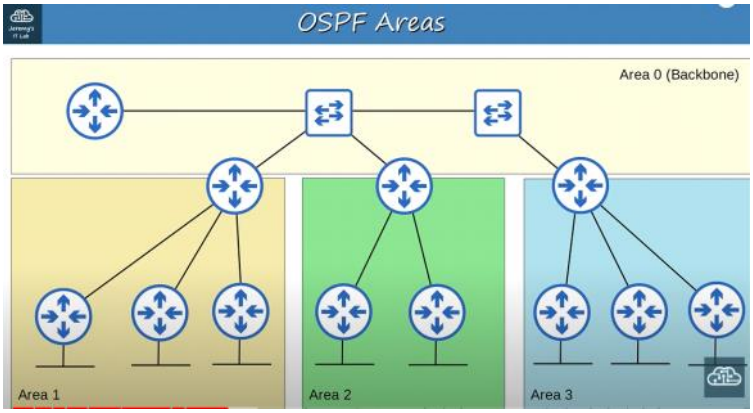
- In OSPF, there are three main steps in the process of sharing LSAs and determining the best route to each destination in the network.

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

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

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

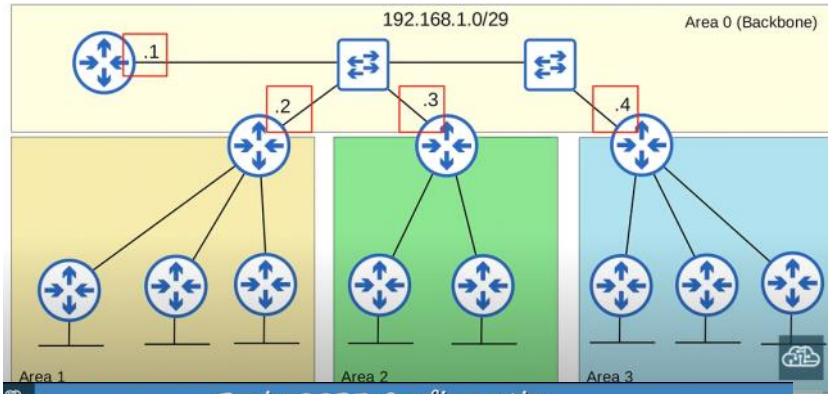
- OSPF uses **areas** to divide up the network.
- Small networks can be *single-area* without any negative effects on performance.
- In larger networks, a single-area design can have negative effects:
  -the SPF algorithm takes more time to calculate routes
  -the SPF algorithm requires exponentially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

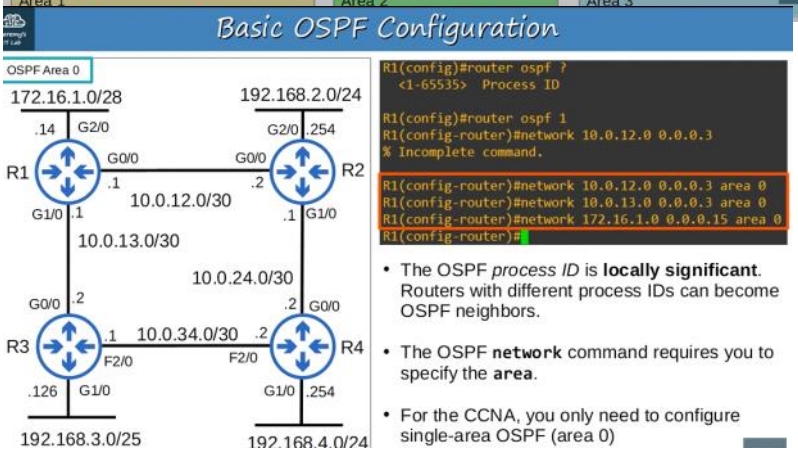


- An **area** is a set of routers and links that share the same LSDB.
- The **backbone area** (area 0) is an area that all other areas must connect to.
- Routers with all interfaces in the same area are called **internal routers**.
- Routers with interfaces in multiple areas are called **area border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

ABRs maintain a separate LSDB for each area they are connected to. It is recommend that you connect an ABR to a maximum of 2 areas. Connecting an ABR to 3+ areas can overburden the router.

- OSPF areas should be *contiguous*.
- 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.






```
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**.
- 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
```

- You already know this command from RIP and EIGRP.
- The **passive-interface** command tells the router to stop sending OSPF 'hello' messages out of the interface.
- However, the router will continue to send LSAs informing it's neighbors about the subnet configured on the interface.
- You should always use this command on interfaces which don't have any OSPF neighbors.

default-information originate - command that advertises the default route configured in R1 to other routers

R1(config-router)# router-id 1.1.1.1
(if any warning stated run **clear ip ospf process** to clear the current one and add this one)

show ip protocols

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

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

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

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

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

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

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

1. Configure the appropriate hostnames and IP addresses on each device. Enable router interfaces (You don't have to configure ISPR1)
2. Configure a loopback interface on each router (1.1.1.1/32 for R1, 2.2.2.2/32 for R2, etc.)
3. Configure OSPF on each router.
   Enable OSPF on each interface (including loopback interfaces).
   (Do not enable OSPF on R1's Internet link)
   Configure passive interfaces where appropriate (including loopback interfaces).
4. Configure R1 as an ASBR that advertises a default route in to the OSPF domain.
5. Check the routing tables of R2, R3, and R4. What default route(s) were added?

do show ip interface brief (shows not the subnet masks)
do show int l0 (shows in detail)

show ip ospf database (to see all the link states)
show ip ospf neighbor

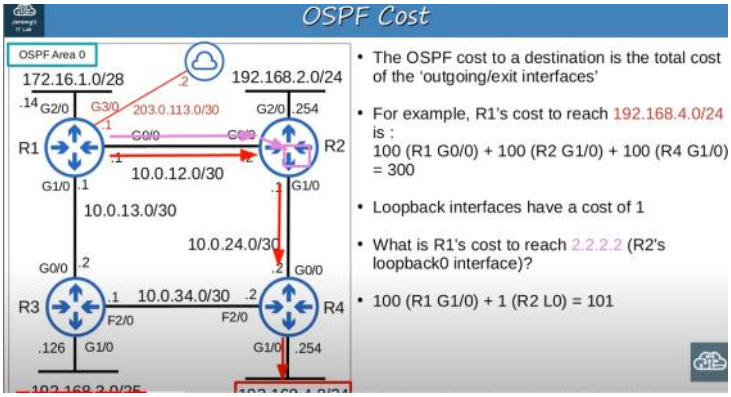
show ip ospf interface

OSPF Cost

- OSPF's metric is called **cost**
- It is automatically calculated based on the bandwidth (speed) of the interface.
- It is calculated by dividing a **reference bandwidth** value by the interface's bandwidth.
- The default reference bandwidth is 100 mbps.
  Reference: 100 mbps / **Interface**: 10 mbps = cost of 10
  Reference: 100 mbps / **Interface**: 100 mbps = cost of 1
  Reference: 100 mbps / **Interface**: 1000 mbps = cost of 1??
  Reference: 100 mbps / **Interface**: 10000 mbps = cost of 1??
- All values less than 1 will be converted to 1.
- Therefore FastEthernet, Gigabit Ethernet, 10Gig Ethernet, etc. are equal and all have a cost of 1 by default.

- You can (and should!) change the reference bandwidth with this command:
  `R1(config-router)# auto-cost reference-bandwidth megabits-per-second`
- The command is entered in megabits per second (default is 100)
- You should configure the same reference bandwidth on all OSPF routers in the network.
- 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)

OSPF Cost



- 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

R1(config-if)# ip ospf cost 10000

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

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

- Three ways to modify the OSPF cost:

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

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

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

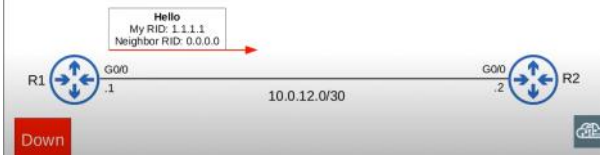
#show ip ospf int brief

OSPF Neighbors

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

OSPF Neighbors – Down State

- OSPF is activated on R1's 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**.



OSPF Neighbors – 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

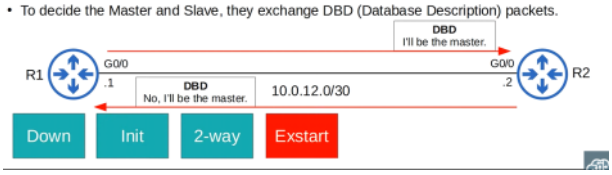
OSPF Neighbors – 2-way State

- 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.
  Now both routers are 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.
  (I will talk about OSPF network types and DR/BDR elections in Day 28)

OSPF Neighbors – 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.



OSPF Neighbors – 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.

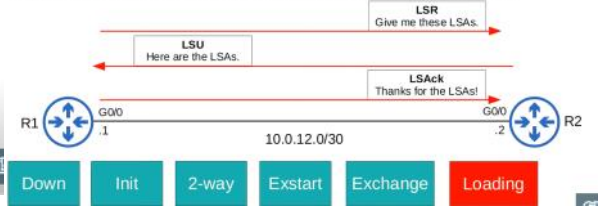
The routers compare the information in the DBD they received to the information in their own LSDB to determine which LSAs they must receive from their neighbor.

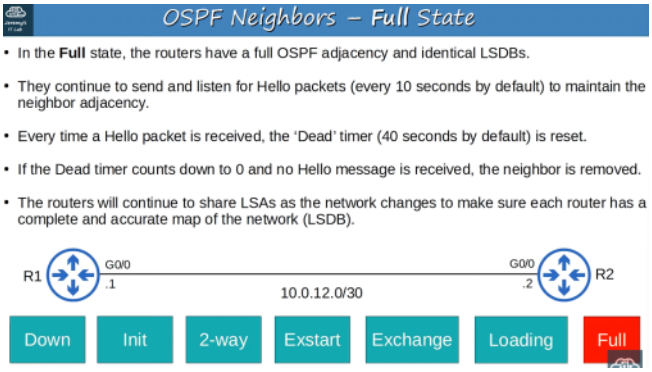
OSPF Neighbors – Loading State

In the **Loading** state, routers send Link State Request (LSR) messages to request that their neighbors send them any LSAs they don't have.

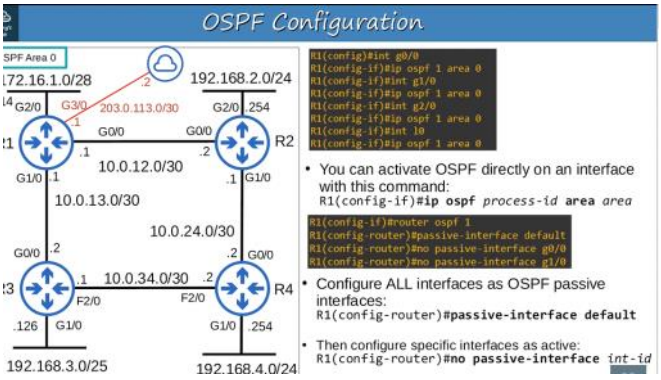
LSAs are sent in Link State Update (LSU) messages.

The routers send LSAck messages to acknowledge that they received the LSAs.





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.



Default timer for hello,dead are 10,40 in seconds

Area can also be writtern in dotted decimal or a single digit wither way works

Protocol number is hexadecimal 89 in ip

If we configure the loopback interfaces:

- When the interface(router's) for which the destination is in the source fails the source cant reach router if loopback interface is not set up
- But if setup it will go through another route and go through the other interface of that router as destination as its loopback interface

OSPF Network Types

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

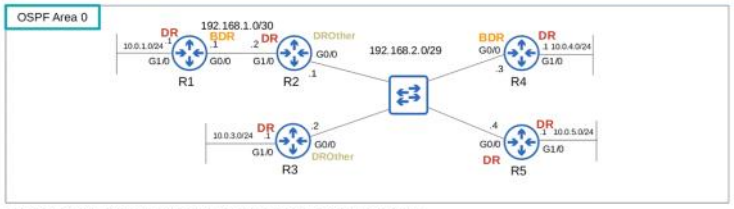
Broadcast
-enabled by default on **Ethernet** and **FDDI** (Fiber Distributed Data Interfaces) interfaces

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

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

3.4	Configure and verify single area OSPFv2
3.4.a	Neighbor adjacencies
3.4.b	Point-to-point
3.4.c	Broadcast (DR/BDR selection)
3.4.d	Router ID

OSPF Broadcast Network Type



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

- 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' becomes the BDR
- The default OSPF interface priority is 1 on all interfaces

```
R5#show ip ospf interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 192.168.2.4/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 5.5.5.5, Network Type BROADCAST, Cost: 1
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           1         no          no            Base
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 5.5.5.5, Interface address 192.168.2.4
  Backup Designated router (ID) 4.4.4.4, Interface address 192.168.2.3
```

R2(config)# int g0/0
R2(config-if)# ip ospf priority 255

If you set the OSPF interface priority to 0, the router CANNOT be the DR/BDR for the subnet.

The DR/BDR election is 'non-preemptive'. Once the DR/BDR are selected they will keep their role until OSPF is reset, the interface fails/is shut down, etc.

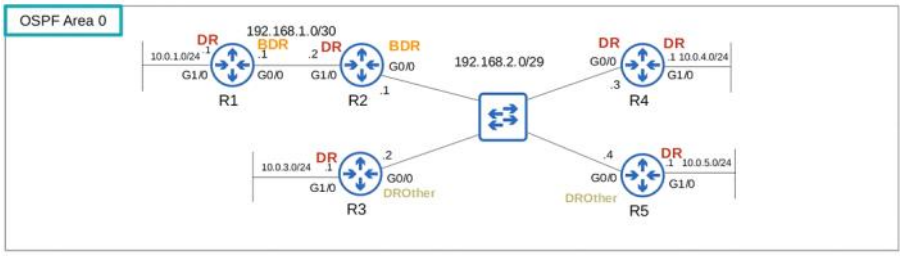
R5# clear ip ospf process

- **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 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 using address 224.0.0.6

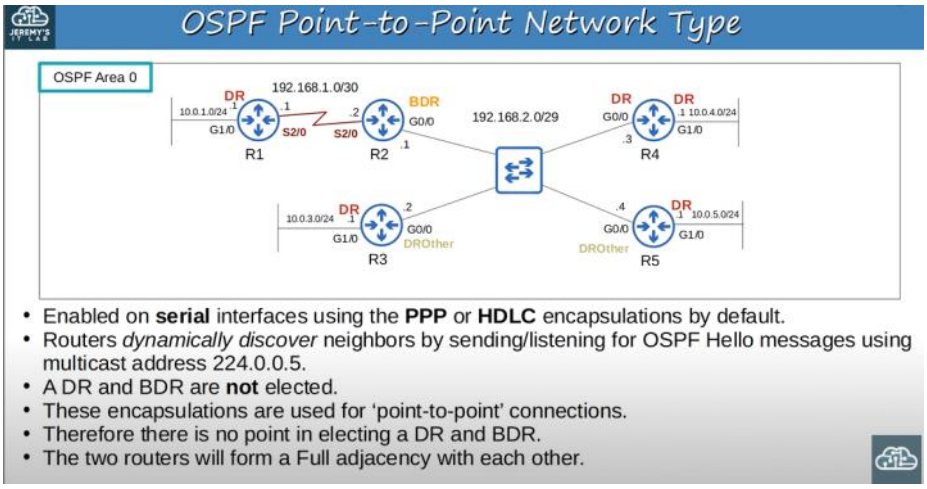
The DR and BDR will form a FULL adjacency with ALL routers in the subnet.
DROthers will form a FULL adjacency only with the DR/BDR.

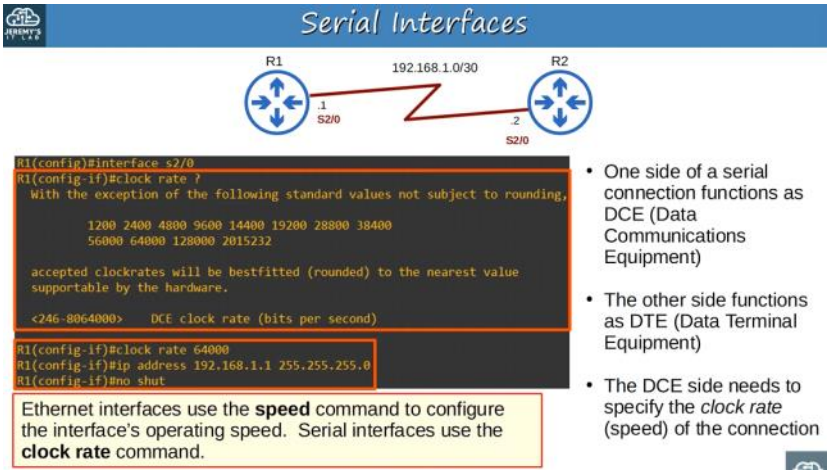


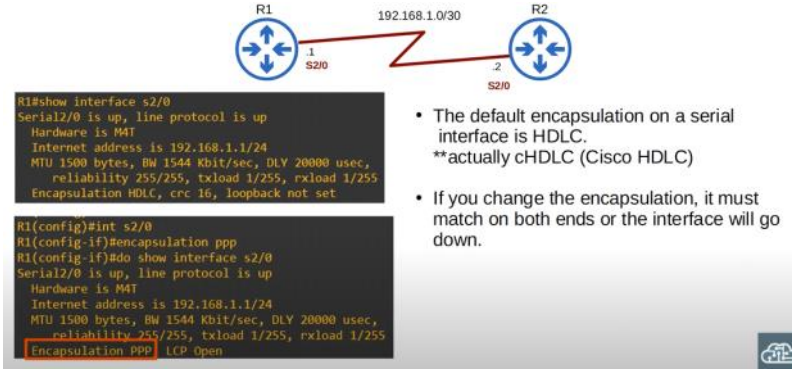

```
R3#show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Gi0/0        1     0               192.168.2.2/29     1     DROTH 2/3
Gi1/0        1     0               10.0.3.1/24        1     DR    0/0
```

F/C = Full adjacency/Total Count of neighbours

PPP,HDLC are layer encapsulations (like ethernet but here serial cable)







R1# show controllers s2/0 - (to see if this is dte/dce)

Instead of DR,BDR,DROTHER for a point to point connection - will be displayed

R1(config-if)# ip ospf network networktype(broadcast/non-broadcast/point-to-multipoint/point-to-point)

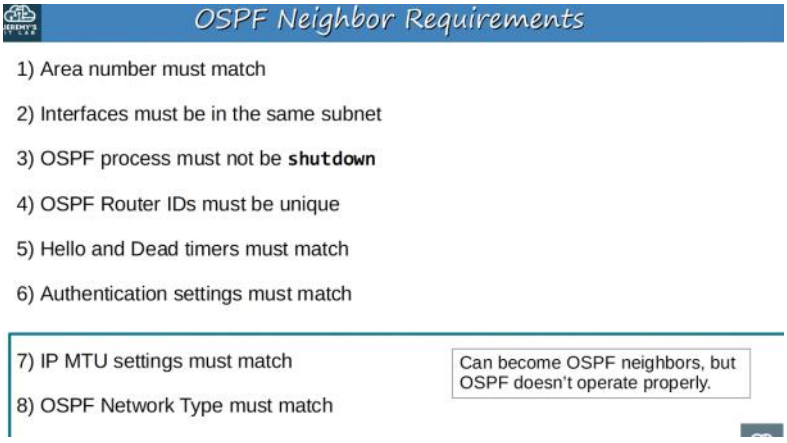
- NOTE: Not all network types work on all link types (for example, a serial link cannot use the broadcast network type)

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

To manually configure timers:
R2(config-if)# ip ospf hello-interval 5
R2(config-if)# ip ospf dead-interval 20
R2(config-if)# no ip ospf hello-interval/dead-interval (for having the default values)

To configure authentication:
R2(config-if)# ip ospf authentication-key jeremy
R2(config-if)# ip ospf authentication
(configure in all routers)

IP MTU(default 1500)(maximum size of an ip packet)(maximum transferrable units)
R2(config-if)# ip mtu 1400
R2(config-if)# no ip mtu (to set default value)



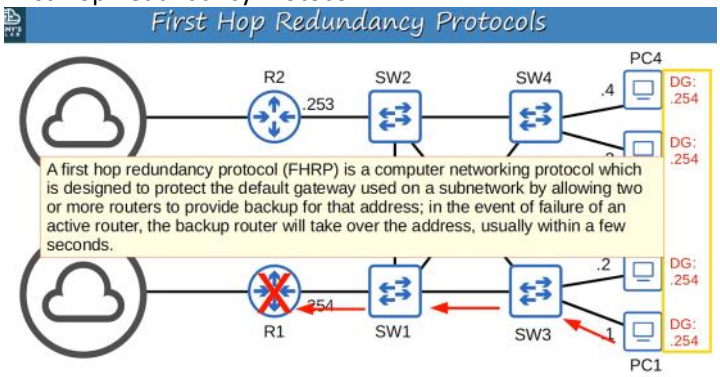
If ospf network type does not match then the neighbour adjacency will be formed but the routes will not be written in the routing table

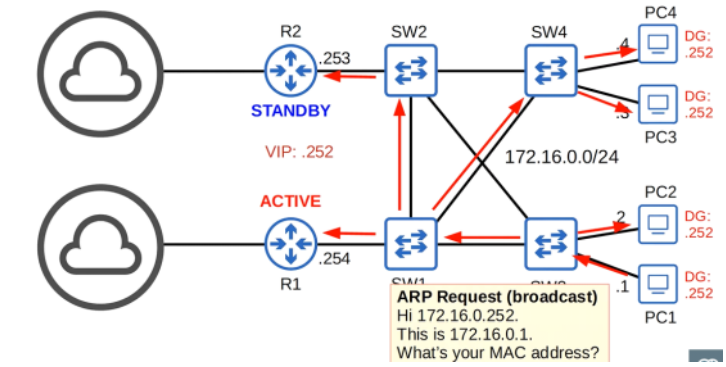
- The OSPF LSDB is made up of LSAs.
- There are 11 types of LSA, but there are only 3 you should be aware of for the CCNA:
  Type 1 (Router LSA)
  Type 2 (Network LSA)
  Type 5 (AS External LSA)
- **Type 1 (Router LSA)**
  -Every OSPF router generates this type of LSA.
  -It identifies the router using its router ID.
  -It also lists networks attached to the router's OSPF-activated interfaces.
- **Type 2 (Network LSA)**
  -Generated by the DR of each 'multi-access' network (ie. the **broadcast** network type).
  -Lists the routers which are attached to the multi-access network.
- **Type 5 (AS-External LSA)**
  -Generated by ASBRs to describe routes to destinations outside of the AS (OSPF domain).

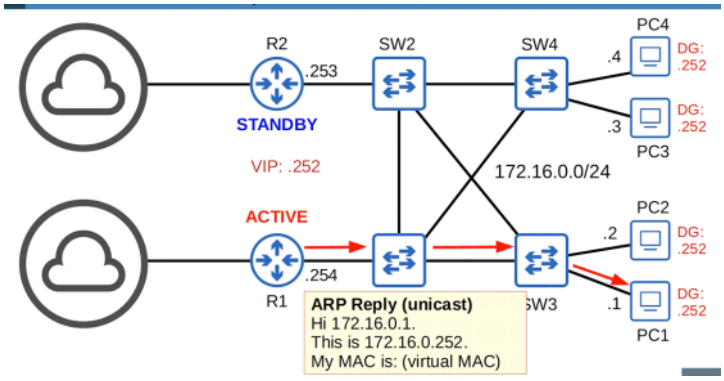
CONCEPTS PART6

Sunday, February 12, 2023 3:35 PM

First Hop Redundancy Protocol

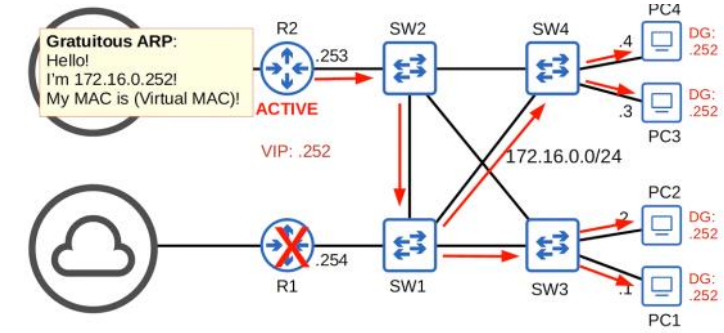






Gratuitous ARP: ARP replies sent without being requested
(no ARP request message was received).

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



FHRPs 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 its active

FHRPs 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 its active role automatically

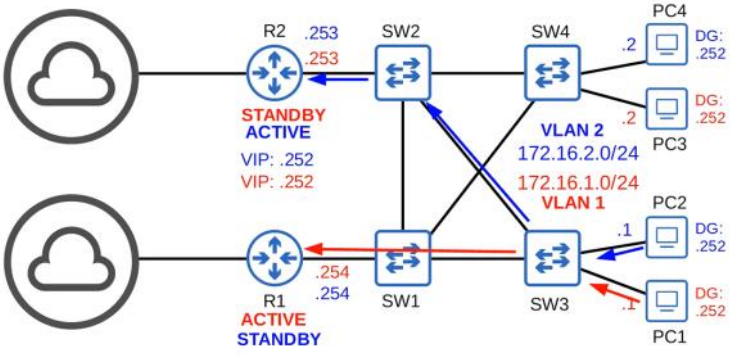
First Hop Redundancy Protocols

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

HSRP (Hot Standby Router Protocol)

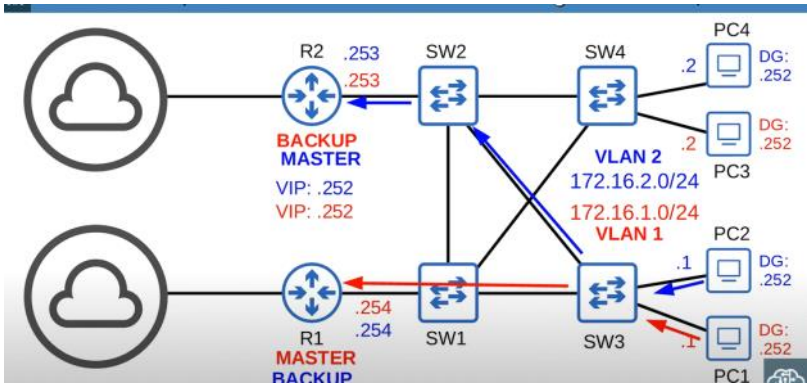
- Cisco proprietary.
- An **active** and **standby** router are elected.
- There are two versions: **version 1** and **version 2**.
  Version 2 adds IPv6 support and increases the number of *groups* that can be configured.
- Multicast IPv4 address: v1 = 224.0.0.2
  v2 = 224.0.0.102
  `0000.0c07.ac01`
- Virtual MAC address: v1 = 0000.0c07.ac**XX** (**XX** = HSRP group number)
  v2 = 0000.0c9f.f**XXX** (**XXX** = HSRP group number)
  `0000.0c9f.f001`
- In a situation with multiple subnets/VLANs, you can configure a different active router in each subnet/VLAN to load balance.

HSRP (Hot Standby Router Protocol)

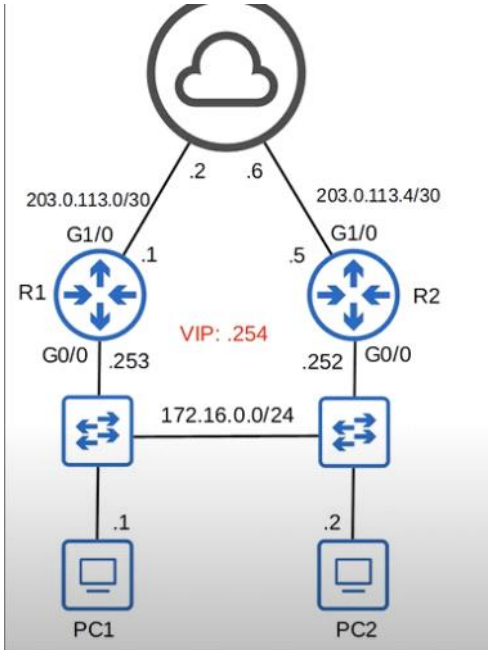


VRRP (Virtual Router Redundancy Protocol)

- Open standard
- A **master** and **backup** router are elected.
- Multicast IPv4 address: 224.0.0.18
- Virtual MAC address: 0000.5e00.01**XX** (**XX** = VRRP group number)
  `0000.5e00.01c8` (0xc8 = 200)
- In a situation with multiple subnets/VLANs, you can configure a different master router in each subnet/VLAN to load balance.



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



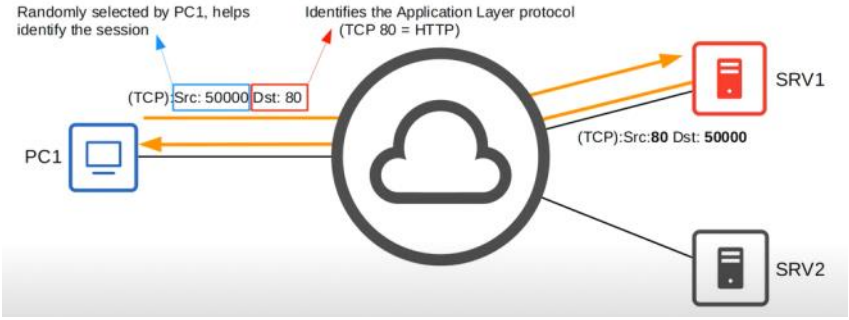
R1(config)# interface g0/0
R1(config-if)# standby version 2 (if wanted to use version 2)
R1(config-if)# standby 1 ip 172.16.0.254
R1(config-if)# standby 1 priority 200
R1(config-if)# standby 1 preempt (only necessary on the router you want to become active when coming online)
R2(config-if)# standby version 2 (if wanted to use version 2)
R2(config-if)# standby 1 ip 172.16.0.254

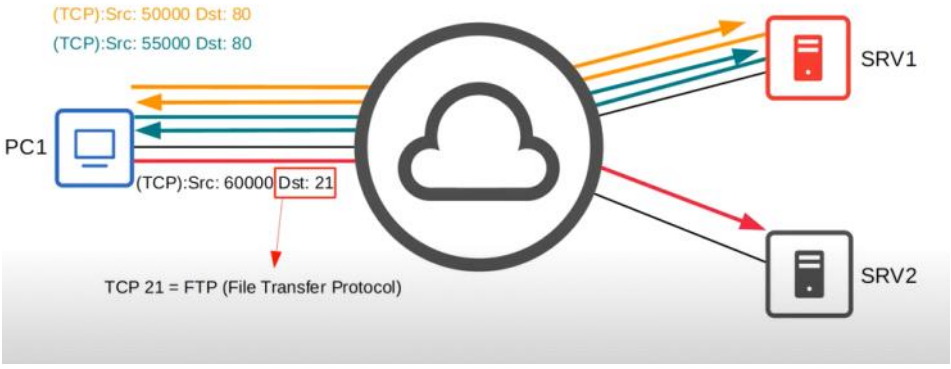
The **active router** is determined in this order:
1 – Highest priority (default 100)
2 – Highest IP address
Preempt causes the router to take the role of active router, even if another router already has the role.

HSRP version 1 and version 2 are not compatible.
If R1 uses version 2, R2 must use version 2 also.

R1# show standby (to see the config of the router for hsrp)

A session is an exchange of data between two or more communicating devices.





Functions of Layer 4 (Transport Layer)

- Provides transparent transfer of data between end hosts.
- Provides (or doesn't provide) various services to applications:
  - → reliable data transfer
  - → error recovery
  - → data sequencing
  - → flow control
- Provides Layer 4 addressing (**port** numbers).

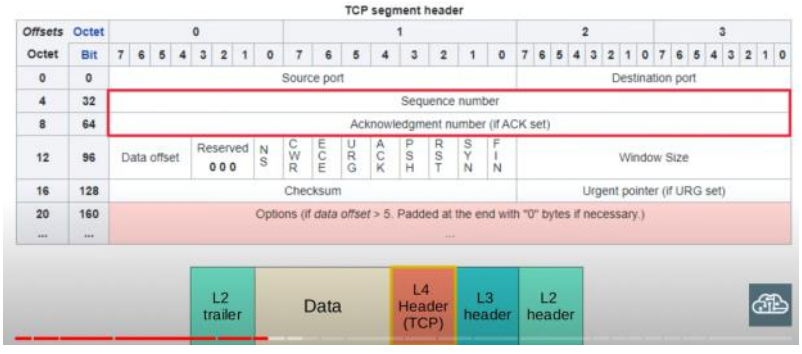
  ↳Identify the Application Layer protocol
  ↳Provides session multiplexing.

  ↳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

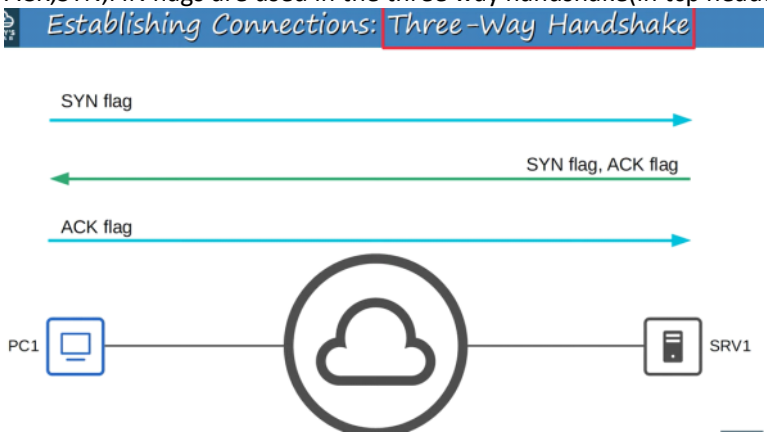
TCP (Transmission Control Protocol)

- TCP is connection-oriented.
  ↳ Before actually sending data to the destination host, the two hosts communicate to establish a connection. Once the connection is established, the data exchange begins.
- TCP provides reliable communication.
  ↳ The destination host must acknowledge that it received each TCP segment.
  ↳ 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.

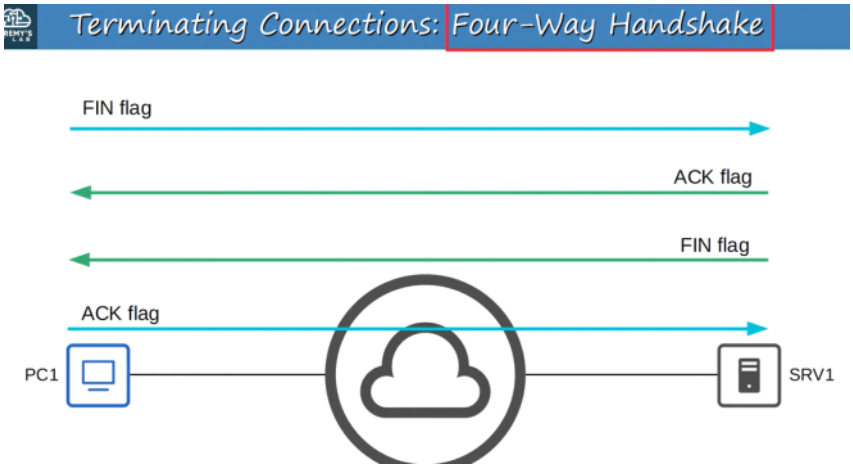
TCP Header

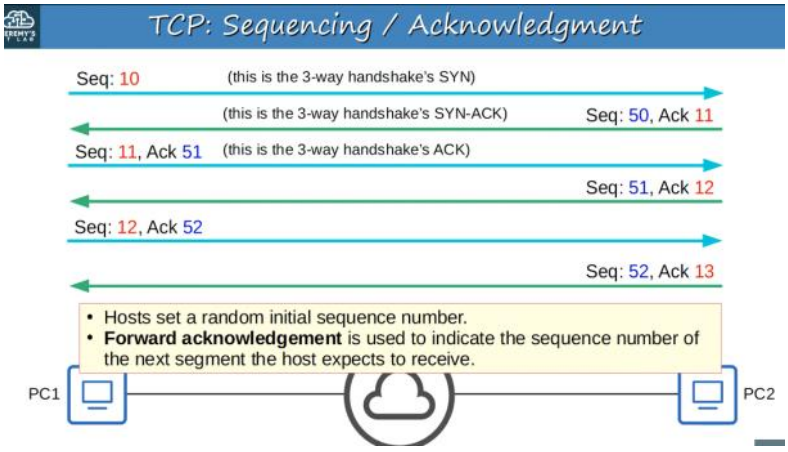


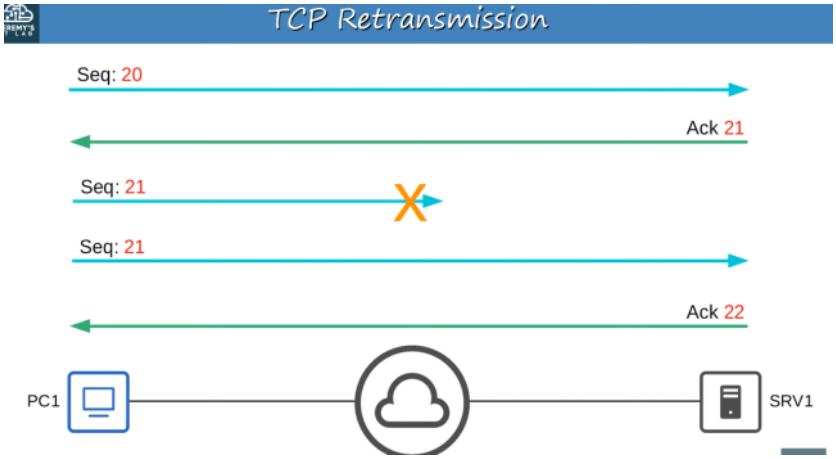
Window size is used for flow control
ACK,SYN,FIN flags are used in the three way handshake(in tcp header)



ACK,FIN are used in terminating connections(4 way handshake)







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 is **not** connection-oriented.
  ↳ 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, acknowledgments are not sent for received segments. If a segment is lost, UDP has no mechanism to re-transmit 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 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															

Comparing TCP & UDP

- TCP provides more features than UDP, but at the 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 & 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

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)

What about IPv5?

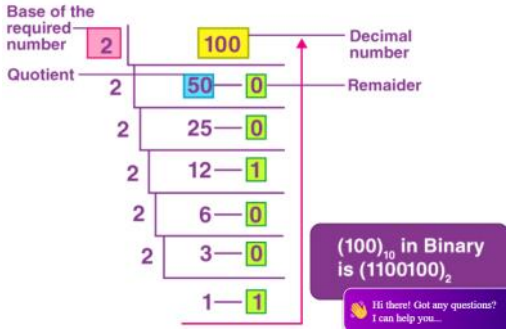
- 'Internet Stream Protocol' was developed in the late 1970s, but never actually introduced for public use.
- It was never called 'IPv5', but it used a value of 5 in the Version field of the IP header.
- So, when the successor to IPv4 was being developed, it was named IPv6.

Hexadecimal

- Binary / Base 2 / 0b
  0, 1
- 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

10 ← Is that decimal 10? Or binary 10 (=decimal 2)? Or hexadecimal 10 (=decimal 16)?

0b10



Binary → Hexadecimal 1

0b11011011 = 0x??

0b1101 0b1011 — Split the number into 4-bit groups

0d13 0d11 — Convert each 4-bit group to decimal

0xD 0xB — Convert each decimal number to hexadecimal

0b11011011 = 0xDB — That's the answer

Hexadecimal → Binary 1

0xEC = 0b??

0xE 0xC — Split up the hexadecimal digits

0d14 0d12 — Convert each hexadecimal digit to decimal

0b1110 0b1100 — Convert each decimal number to binary

0xEC = 0b11101100 — That's the answer

Why IPv6?

- The main reason is that **there simply aren't enough IPv4 address available!**
- There are 4,294,967,296 (2^{32}) IPv4 addresses available.
- When IPv4 was being designed 30 years ago, the creators had no idea the Internet would be as large as it is today.
- VLSM, private IPv4 addresses, and NAT have been used to conserve the use of IPv4 address space.
- Those are short-term solutions.
- The long-term solution is IPv6.

- IPv4 address assignments are controlled by IANA (Internet Assigned Numbers Authority)
- IANA distributes IPv4 address space to various RIRs (Regional Internet Registries), which then assign them to companies that need them.

On 24 September 2015 ARIN declared exhaustion of the ARIN IPv4 addresses pool.

On 21 August 2020, LACNIC announced that it had made its final IPv4 allocation.

- An IPv6 address is **128 bits**.
- 4*the bits of an IPv4 address = 4*the number of possible addresses? NO
- Every additional bit **doubles** the number of possible addresses.
- There are 340,282,366,920,938,463,463,374,607,431,768,211,456 IPv6 addresses.
  There are .....4,294,967,296 IPv4 addresses.
- Example IPv6 address in binary:
  0010000000000001000011011011100001011001000101111110101010111101011001010110001000010111111010101100100100101101010110011011110 1

  ↪~~32.1.13.184.89.23.234.189.101.98.23.234.201.45.89.189~~

  ↪2001:0DB8:5917:EABD:6562:17EA:C92D:59BD
  1 2 3 4 5 6 7 8

In ipv6 there is not subnet mask and /26 or prefix is used

- **Leading 0s** can be removed
  2001:0DB8:000A:001B:20A1:0020:0080:34BD

  ↓

  2001:DB8:A:1B:20A1:20:80:34BD

- **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

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 shortened IPv6 addresses

- Put leading 0s where needed (all quartets should have 4 hexadecimal characters)

FE80::2:0:0:FBE8

↓

FE80::0002:0000:0000:FBE8

- If a double colon is used, replace it with all-0 quartets. Make sure there are 8 quartets in tot

FE80::0002:0000:0000:FBE8 5 quartets (8 quartets, but only 5 are written)

↓

FE80:0000:0000:0000:0002:0000:0000:FBE8 8 quartets

Finding the IPv6 prefix (global unicast addresses)

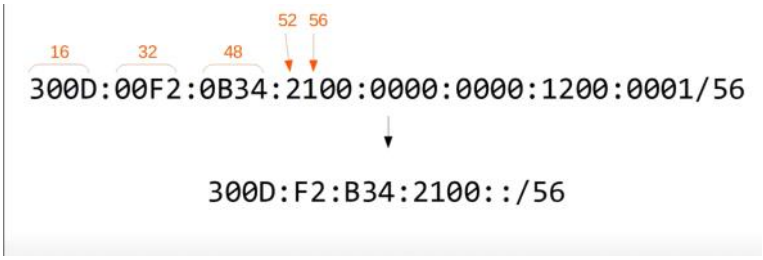
- Typically, an enterprise requesting IPv6 addresses from their ISP will receive a /48 block.
- Typically, IPv6 subnets use a /64 prefix length.
- That means an enterprise has 16 bits to use to make subnets.
- The remaining 64 bits can be used for hosts.

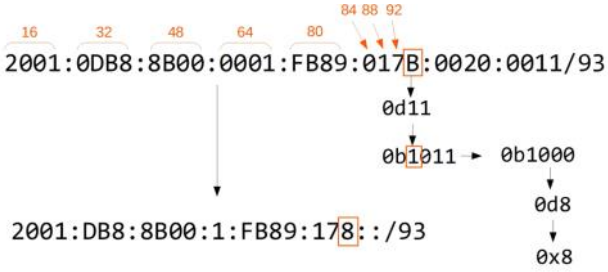
2001:0DB8:8B00:0001:0000:0000:0000:0001/64

48-bit 'global routing prefix' assigned by the ISP

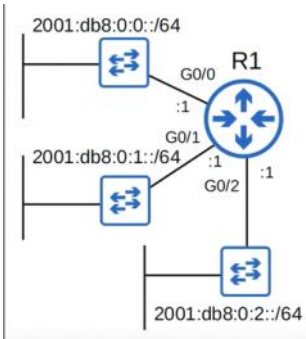
16-bit 'subnet identifier', used by the enterprise to make various subnets

64-bit 'interface identifier', the host portion of the address





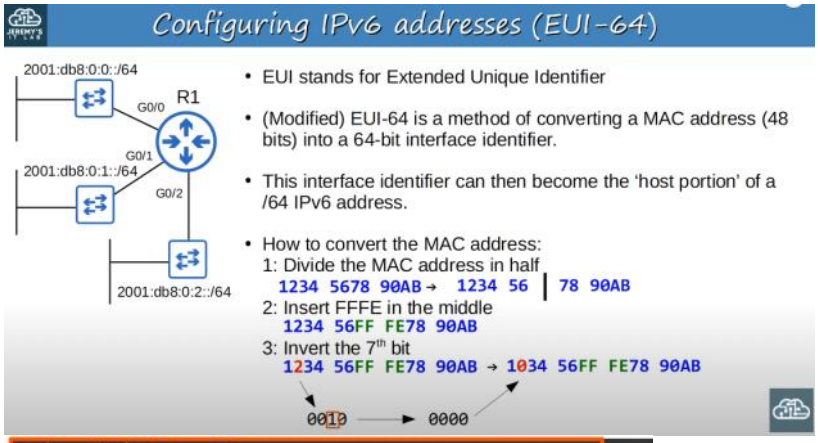
Host Address	Prefix
FE80:0000:0000:0000:4c2c:e2ed:6a89:2a27/9	FE80::/9
2001:0DB8:0001:0B23:BA89:0020:0000:00C1/64	2001:DB8:1:B23::/64
2001:0DB8:0BAD:CAFE:1300:0689:9000:0CDF/71	2001:DB8:BAD:CAFE:1200::/71
2001:0DB8:0000:FEED:0DAD:018F:6001:0DA3/62	2001:DB8:0:FEEC::/62
2001:0DB8:9BAD:BABE:0DE8:AB78:2301:0010/63	2001:DB8:9BAD:BABE::/63




```
R1(config)#
R1(config)#ipv6 unicast-routing
R1(config)#
R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8:0:0::1/64
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#int g0/1
R1(config-if)#ipv6 address 2001:db8:0:1::1/64
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#int g0/2
R1(config-if)#ipv6 address 2001:0db8:0000:0002:0000:0000:0000:0001/64
R1(config-if)#no shutdown
R1(config-if)#
```

```
R1#show ipv6 interface brief
GigabitEthernet0/0     [up/up]
    FE80::EF8:22FF:FE36:8500
    2001:DB8::1
GigabitEthernet0/1     [up/up]
    FE80::EF8:22FF:FE36:8501
    2001:DB8:0:1::1
GigabitEthernet0/2     [up/up]
    FE80::EF8:22FF:FE36:8502
    2001:DB8:0:2::1
GigabitEthernet0/3     [administratively down/down]
    unassigned
R1#
```

For ipv6 link-local addresses are automatically created but for ipv4 we need to create it since it is not by default configured




```
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
```

Why invert the 7th bit?

- MAC addresses can be divided into two types:
  - → **UAA** (Universally Administered Address)
    - ↳ Uniquely assigned to the device by 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 identify 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**

Global unicast addresses

- **Global unicast** IPv6 addresses are public addresses which can be used over the Internet.
- Must register to use them. Because they are public addresses, it is expected that they are 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 aren't reserved for other purposes.

`2001:0DB8:8B00:0001:0000:0000:0000:0001/64`

- 48-bit 'global routing prefix' assigned by the ISP
- 16-bit 'subnet identifier', used by the enterprise to make various subnets
- 64-bit 'interface identifier', the host portion of the address

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. They can be used freely within internal networks and don't need to be globally unique (*). Can't be routed over the Internet.
- Uses the address block `FC00::/7` (`FC00::` to `FDFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF`)
- However, a later update requires 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.

`FD45:93AC:8A8F:0001:0000:0000:0000:0001/64`

- Indicates a unique local address
- 64-bit 'interface identifier', the

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. They can be used freely within internal networks and don't need to be globally unique (*). Can't be routed over the Internet.
- Uses the address block `FC00::/7` (`FC00::` to `FDFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF`)
- However, a later update requires 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.

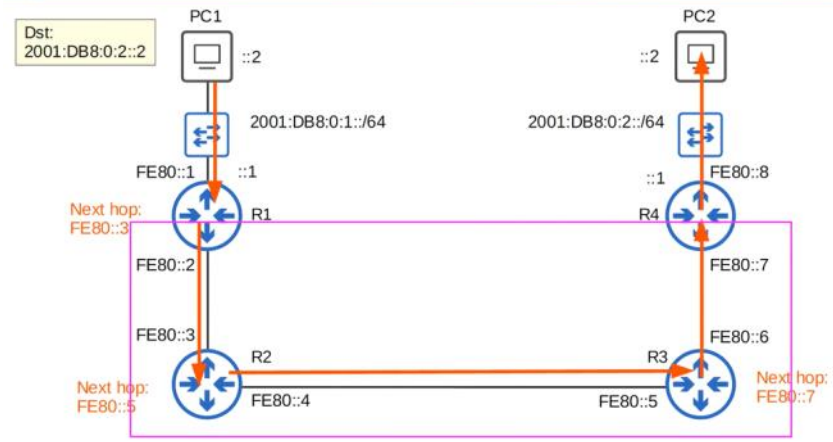
`FD45:93AC:8A8F:0001:0000:0000:0000:0001/64`

- `FD` — Indicates a unique local address
- `45:93AC:8A8F` — 40-bit 'global ID', which should be randomly generated
- `0001` — 16-bit 'subnet identifier', used by the enterprise to make various subnets
- `0000:0000:0000:0001` — 64-bit 'interface identifier', the host portion of the address

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`)
- However, the standard states that the 54 bits after `FE80/10` should be all 0, 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). Routers **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 neighbor adjacencies)
  - → next-hop addresses for static routes
  - → *Neighbor Discovery Protocol* (NDP, IPv6's replacement for ARP) uses link-local addresses to function

Link-local addresses



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!)

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 addresses in the previous slide all use the 'link-local' scope (FF02), which stays in the local subnet.
- IPv6 multicast scopes:
  ↳ **Interface-local** (FF01): The packet doesn't leave the local device. Can be used to send traffic to a service within the local device.

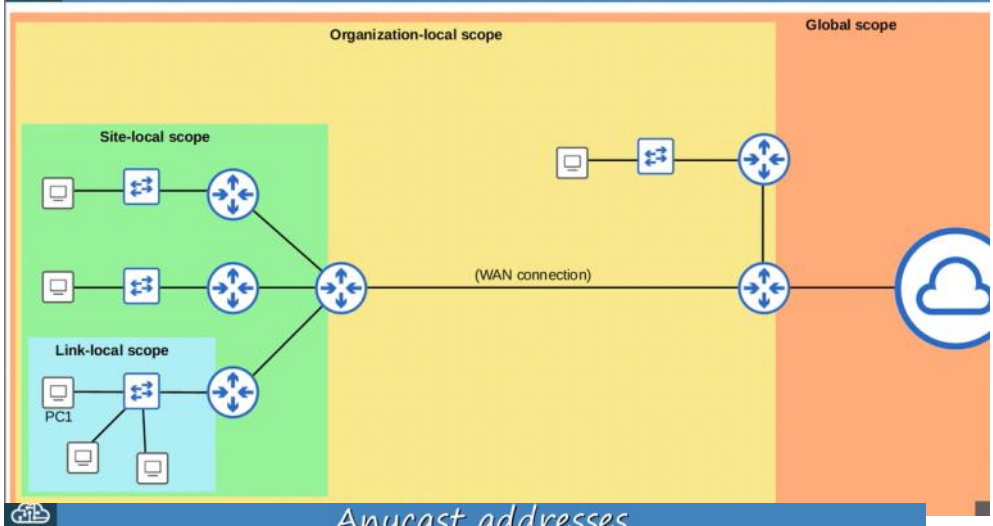
  ↳ **Link-local** (FF02): The packet remains in the local subnet. Routers will not route the packet between subnets.

  ↳ **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).

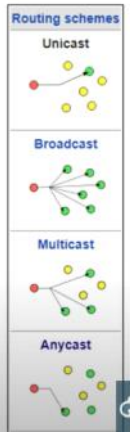
↳ **Global** (FF0E): No boundaries. Possible to be routed over the Internet.

Multicast address scopes



Anycast addresses

- **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 address (global unicast, unique local) and specify it as an anycast address:
  R1(config-if)# **ipv6 address 2001:db8:1:1::99/128 anycast**



- `::` = 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`
- `::1` = 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

R1's G0/1 interface has a MAC address of `0D2A.4FA3.00B1`.
What will G0/1's IPv6 address be after issuing the following command?
R1(config-if)# **ipv6 address 2001:db8:0:1::/64 eui-64**

a) `2001:db8:0:1:0B2A:4FFF:FFA3:B1`

b) `2001:db8:0:1:C2A:4FFF:FEA3:B1`

c) `2001:db8:0:1:0F2A:4FFF:FFA3:B1`

d) `2001:db8:0:1:F2A:4FFF:FEA3:B1`

To give the router's link local address we need to specify the interface and the ipv6 address in order to route them up

IPv6 Address Representation

- An RFC (Request for Comments) is a publication from the ISOC (Internet Society) and associated organizations like the IETF (Internet Engineering Task Force), and are the official documents of Internet specifications, protocols, procedures, etc.
- RFC 5952 is '**A Recommendation for IPv6 Address Text Representation**'
- Before this RFC, IPv6 address representation was more flexible
  - → You could remove leading 0s, or leave them
  - → You could replace all-0 quartets with ::, or leave them
  - → You could use upper-case 0xA,B,C,D,E,F, or lower-case 0xa,b,c,d,e,f
- RFC 5952 suggests standardizing IPv6 address representation

- Leading 0s MUST be removed.
  `2001:0db8:0000:0001:0f2a:4fff:fea3:00b1`
  → `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 '::')
  `2001:0000:0000:0000:0f2a:0000:0000:00b1`
  → `2001::f2a:0:0:b1`
- If there are two equal-length choices for the ::, use :: to shorten the one on the left.
  `2001:0db8:0000:0000:0f2a:0000:0000:00b1`
  → `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

Verison:

- 4 bits in length
- Always set to 0b0110 to indicate ipv6

Traffic class:

- 8 bits in length
- Used for qos

Flow label:

- 20 bits in length
- Used to identify specific traffic flows

Payload length:

- Length:16bits
- Indicated the length of the payload

Next header:

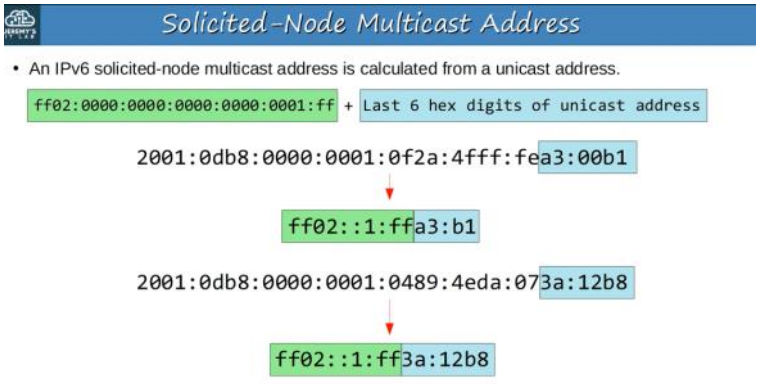
- 8 bits in lengthsame function of protocol field in ipv4

Hop limit:

- 8 bits in length
- Value if decremented by 1 by each router that forwards it and at last when it is 0 packet is discarded
- Same as ipv4 ttl

Source / destination:

- 128 bits each




```
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

- 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-node multicast addresses to learn the MAC address of other hosts.
  *(ARP in IPv4 uses broadcast messages)
- Two message types are used:
  1) Neighbor Solicitation (NS) = ICMPv6 Type 135
  2) Neighbor Advertisement (NA) = ICMPv6 Type 136

Neighbor Solicitation (NS)

Hi, what's your MAC address?

- **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)

Hi, my MAC address is ca02.097c.0008.

- **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
```

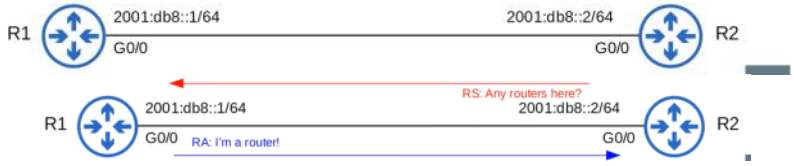
R1# show ipv6 neighbor

Neighbor Discovery Protocol

- Another function of NDP allows hosts to automatically discover routers on the local network.
- Two messages are used for this process:

1) 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.

2) 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. `2001: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]
```

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 gets a reply, it means another host on the network is already using the address.

```
*Oct 31 11:28:48.318: %IPV6_ND-4-DUPLICATE: Duplicate address 2001:DB8::1 on GigabitEthernet0/0
```

- 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.

IPv6 Static Routing

`ipv6 route` *`destination/prefix-length {next-hop | exit-interface [next-hop]} [ad]`*

Directly attached static route: Only the exit interface is specified.

`ipv6 route` *`destination/prefix-length exit-interface`*

~~`R1(config)# ipv6 route 2001:db8:0:3::/64 g0/0`~~

In IPv6, you CAN'T 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`*

```
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`*

```
R1(config)# ipv6 route 2001:db8:0:3::/64 g0/0 2001:db8:0:12::2
```

IPv6 Static Routing

`ipv6 route` *`destination/prefix-length {next-hop | exit-interface [next-hop]} [ad]`*

Network route:

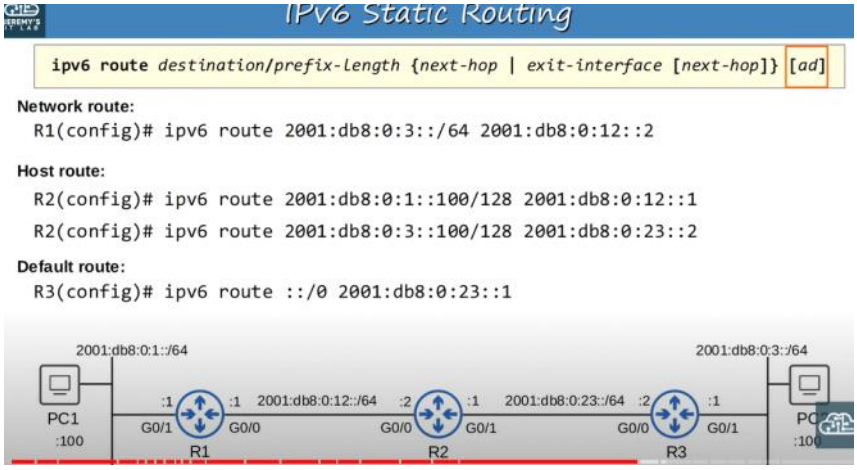
```
R1(config)# ipv6 route 2001:db8:0:3::/64 2001:db8:0:12::2
```

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
```

Default route:

```
R3(config)# ipv6 route ::/0 2001:db8:0:23::1
```



When we enable ipv6 unicast routing in all the routers and if we enable automatic ipv6 gateway address it will pick from the router nearest , it's ipv6 baddress,link local address will be learned from the corresponding router

When configuring routes for ipv6 we need to specify higher AD than 1 so that the particular route will be in non priority

In router give the ipv6 address with prefix and eui-64 , and ipv6 enable
And if multiple lan's give unicast routing and configure the static routes correctly and in each pc give correct address of router as default gateway